2006-12

High-fidelity real-time trajectory optimization for reusable launch vehicles

Bollino, Kevin P.

Monterey, California.: Naval Postgraduate School, 2006.

http://hdl.handle.net/10945/10056

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

HIGH-FIDELITY REAL-TIME TRAJECTORY OPTIMIZATION FOR REUSABLE LAUNCH VEHICLES

by

Kevin P. Bollino

December 2006

Dissertation Supervisor: I. Michael Ross

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2006	**3. REPORT TYPE AND DATES COVERED** Doctoral Dissertation

4. TITLE AND SUBTITLE: High-Fidelity Real-Time Trajectory Optimization for Reusable Launch Vehicles	**5. FUNDING NUMBERS**
6. AUTHOR Kevin P. Bollino	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Creating simplicity out of complexity, this research abandons the traditional guidance and control architecture for aerospace vehicles and embraces a revolutionary concept based on the principles of nonlinear optimal control theory. Motivated by the emerging needs of the next generation of reusable space vehicles, an autonomous "integrated" guidance and control system is developed that provides a safe approach to the highly constrained and nonlinear reentry problem. A pseudospectral-based optimal guidance scheme is used to generate high-fidelity, vehicle-tailored solutions to reentry trajectory optimization and guidance problems. To provide an autonomous, onboard capability of satisfying final-approach requirements, a new method is developed that includes an automatic generation of landing constraints given any runway geometry. This unique and simple approach avoids significant complexities arising from previous ideas of trajectory segmentation, trimmed flight, and trajectory tracking schemes. When demonstrating the new ideas, it is shown that the proposed approach can easily compensate for large uncertainties and disturbances consisting of hurricane-force wind gusts. An investigation of these new principles for the complete, nonlinear six degree-of-freedom system dynamics indicates that while the results are quite promising, a substantial amount of new theoretical and computational problems remain open, particularly in the area of over-actuated dynamical systems.

14. SUBJECT TERMS Optimal Guidance and Control, Nonlinear Trajectory Optimization, Reentry, X-33, Reusable Launch Vehicles, Path Planning, Replanning, Retargeting, High-Fidelity Modeling, Aerospace Flight Controls, Pseudospectral Methods, DIDO, Real-Time Optimal Control, Pseudospectral Feedback Control, Autonomous Vehicles, Intelligent Systems	**15. NUMBER OF PAGES** 445
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

HIGH-FIDELITY REAL-TIME TRAJECTORY OPTIMIZATION FOR REUSABLE LAUNCH VEHICLES

Kevin P. Bollino
Captain, United States Air Force
B.S., Embry-Riddle Aeronautical University, 1997
M.S., University of Dayton, 2000

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2006

Author:
Kevin P. Bollino

Approved by:

I. Michael Ross, Professor
Mechanical and Astronautical Engineering
Dissertation Supervisor

David B. Doman, Senior Aerospace Engineer
Air Force Research Laboratory

Isaac I. Kaminer, Professor
Mechanical and Astronautical Engineering

Wei Kang, Professor
Applied Mathematics

Fotis A. Papoulias, Associate Professor
Mechanical and Astronautical Engineering

Xiaoping Yun, Professor
Electrical and Computer Engineering

Approved by:
Anthony J. Healey, Chair, Department of Mechanical and
Astronautical Engineering

Approved by:
Julie Filizetti, Associate Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Creating simplicity out of complexity, this research abandons the traditional guidance and control architecture for aerospace vehicles and embraces a revolutionary concept based on the principles of nonlinear optimal control theory. Motivated by the emerging needs of the next generation of reusable space vehicles, an autonomous "integrated" guidance and control system is developed that provides a safe approach to the highly constrained and nonlinear reentry problem. A pseudospectral-based optimal guidance scheme is used to generate high-fidelity, vehicle-tailored solutions to reentry trajectory optimization and guidance problems. To provide an autonomous, onboard capability of satisfying final-approach requirements, a new method is developed that includes an automatic generation of landing constraints given any runway geometry. This unique and simple approach avoids significant complexities arising from previous ideas of trajectory segmentation, trimmed flight, and trajectory tracking schemes. When demonstrating the new ideas, it is shown that the proposed approach can easily compensate for large uncertainties and disturbances consisting of hurricane-force wind gusts. An investigation of these new principles for the complete, nonlinear six degree-of-freedom system dynamics indicates that while the results are quite promising, a substantial amount of new theoretical and computational problems remain open, particularly in the area of over-actuated dynamical systems.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVIATIONS AND ACRONYMS

A/L – Approach and Landing (or Approach to Landing)

a^{2}B – Anti-Aliasing Bellman

AFRL – Air Force Research Laboratory

AFWA – Air Force Weather Agency

AG&C – Advanced Guidance and Control

ALFEX – Automatic Landing Flight Experiment

ALIP – Autoland I-Load Program

AoA – Angle-of-Attack

ARDC-59 – Air Research and Development Command

ASTOS – Aerospace Trajectory Optimization Software

ASTP – Advanced Space Transportation Program

ATC – Air Traffic Control

BA – Bank Angle

BAE – Base-Aero-Engine

CC – Complementarity Condition

CEV – Crew Exploration Vehicle

CFD – Computational Fluid Dynamics

CG – Center of Gravity

CM – Center of Mass

CMP – Covector Mapping Principle

COV – Calculus of Variations

CPU – Central Processing Unit

CR – Crossrange

CS – Complementary Slackness

DCM – Direction Cosine Matrix

DF – Dual Feasibility

DI – Dynamic Inversion

DOF – Degree(s)-of-Freedom

DR – Downrange

DU –Distance Unit

EAGLE - Evolved Acceleration Guidance Logic for Entry

ECEF – Earth-Centered Earth-Fixed

ECI – Earth-Centered-Inertial

EGM96 – 1996 Earth Gravity Model

EoM –Equation(s) of Motion

ESA – Energy-State Approximation

FAA – Federal Aviation Administration

FAC – Final Approach Corridor

FCS – Flight Control System

FCS – Future Combat System

FPA – Flight Path Angle

G&C – Guidance and Control

GB – Gigabyte

GESOP – Graphical Environment for Simulation and Optimization

GHz – Giga-Hertz

GNC – Guidance, Navigation, and Control

GS – Glide Slope

HA – Heading Angle

HAC – Heading Alignment Cylinder/Cone

HAVOC – Hypersonic Aircraft Vehicle Optimization Code

HEE – Hamiltonian Evolution Equation

HiFi – High Fidelity

HMC – Hamiltonian Minimization Condition

HVC – Hamiltonian Value Condition

IAG&C – Integrated Advanced Guidance and Control

IC – Initial Condition

ICBM – Inter-Continental Ballistic Missile

IMU – Inertial Measurement Unit

ISS – International Space Station

JAUS - Joint Architecture for Unmanned Systems

KKT – Karush-Kuhn-Tucker

KSC – Kennedy Space Center

L/D – Lift-to-Drag Ratio

LEO – Low Earth Orbit

LGL – Legendre-Gauss-Lobatto

LQR – Linear Quadratic Regulator

LTV – Linear(ized) Time-Varying

M – Mach Number

MAV – Micro Aerial Vehicle

MAVERIC – Marshall Aerospace Vehicle Representation in C

MECO – Main-Engine Cut-Off

MPC – Model Predictive Control

MRAC – Model Reference Adaptive Control

MSBLS – Microwave Scan Beam Landing System

MSFC – Marshall Space Flight Center

MU – Mass Unit

NAL – National Aerospace Laboratory of Japan

NASA – National Air and Space Administration

NASDA – National Space Development Agency of Japan

NED – North East Down

NL OCP – Nonlinear Optimal Control Problem

NLP – Nonlinear Programming

OCP – Optimal Control Problem

ODE – Ordinary Differential Equation

OPTG – Optimum-Path-To-Go

OTIS – Optimal Trajectories by Implicit Simulation

PC – Predictor-Corrector

PF – Primary Feasibility

PI – Proportional-Integral

PID – Proportional-Integral-Derivative

PLP – Piecewise Linear Programming

PNN – Polynomial Neural Network

PS – Pseudospectral

QEG – Quasi-Equilibrium Glide

QEGC – Quasi-Equilibrium Glide Condition

R&D – Research and Development

RAM – Random Access Memory

RCS – Reaction Control System

Re – Reynolds Number

RHC – Receding Horizon Control

RHS – Right-Hand-Side

RLV – Reusable Launch Vehicle

RV – Reentry Vehicle

SaH – Sample-and-Hold

SLI – Space Launch Initiative

SNOPT – Software for Large-Scale Nonlinear Programming

SOCS – Sparse Optimal Control Software

SQP – Sequential Quadratic Programming

STS – Space Transportation System

TACAN – Tactical Air Navigation

TAEM – Terminal Area Energy Management

TC – Transversality Condition

TM – Transformation Matrix

TPBVP – Two-Point Boundary Value Problem

TPS – Thermal Protection System

TTC – Terminal Transversality Condition

TU – Time Unit

UAV – Unmanned Aerial Vehicle

WGS-84 – 1984 World Geodetic System

ZOH – Zero-Order Hold

To my parents,
for their love and support over the years.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First, and foremost, I would like to gratefully thank my family: Keiko, Maya, and Keila, for their patience and sacrifices over the past few years. Without your support I would have never made it this far. Although my ambitions are often clouded by visions of grandeur, a seeking spirit that ceases to seek, is no more.

Secondly, I would like to express my gratitude to my advisor, Dr. Mike Ross, for his “optimal guidance” throughout my research. Your enigmatic persona has helped lead me to the extremal frontier in my quest of academic achievement. Thanks for never being short of ideas — no matter how many ideas I chase down the rabbit hole, there is always a carrot at the end.

To Dr. Pooya Sekhavat for your expertise on a moment’s notice, practicality, motivating attitude, and serving as an excellent research engineer role model and co-advisor. Also, thanks for running a “cool” Guidance, Navigation, and Control Lab that gave me the opportunity to work in an entertaining and interactive environment amongst friends.

To Dr. David Doman and Dr. Mike Oppenheimer of the Air Force Research Laboratory, for providing the X-33 aerodynamic database, your flight dynamics expertise, and the opportunity to work with and learn from two great leaders in the field of aerospace controls.

To Mr. David Brown and Mr. Robert Creasey (NPS Meteorology), for providing weather data instrumental for the wind gust modeling in this work.

To Dr. Masanori Harada, for sharing your unique problem-solving approach and your encouragement when I was preparing for my qualifying exams.

To Dr. Kwang Song, for your inspiration and dharma teachings that helped me awaken to the true simplicity in the illusion of complexity. Your demeanor is contagious.

Finally, I would like to thank the U.S. Air Force, for giving me and my family such an opportunity. Although the debt will be repaid, the professional and personal rewards are worth many years of service.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. OVERVIEW

With ongoing efforts to improve the safety, reliability, and cost of reusable launch vehicle (RLV) systems and their operations along with the more recent focus on a U.S. Space Shuttle replacement following its retirement around 2012, there is a new opportunity to implement advanced, yet simpler and more effective guidance and control (G&C) methods than used in previous reentry technologies. A key objective that is sought in this endeavor as well as almost all other sectors of technology is the requirement for more intelligent systems that can operate autonomously with less human interaction. Of course this requirement has the underlying stipulation that they operate safely, more reliably, and more efficiently than ever before. To accomplish this objective for the case of an RLV, the G&C during ascent, reentry, or landing must operate autonomously and in real-time. The RLV intelligent control must be capable of adapting to rapidly changing circumstances; handling large external disturbances and large parameter uncertainties; re-generating trajectories (i.e., replanning to the original or alternative landing site); and reconfiguring its control effectors in the event of an unforeseen control failure. In practice, these challenges are met by extensive pre-flight planning involving abort planning for a large but finite number of off nominal conditions such as engine failure, control effector failures, or variations in atmospheric conditions. The preflight planning therefore consumes a substantial amount of manpower and is susceptible to delays in launch schedules.

This chapter begins with background of the RLV reentry problem to include a historical overview of reentry vehicles and associated aerospace challenges; an overview of various methods developed to address the reentry problem; the motivation for this particular research work; highlights of specific contributions that this work delivers to the aerospace field; and then concludes with an overview of the proceeding chapters.

B. BACKGROUND

1. The RLV Reentry Problem

a. Historical Context of Reentry Vehicles

Guiding and controlling objects entering the earth's atmosphere [1],[2] became a major concern with the advent of Inter-Continental Ballistic Missile (ICBM) development beginning in 1950 by the major world powers. At the time, rapid response was a key strategic deterrent for an enemy threat such that by 1965 the United States and the Soviet Union could strike almost any part of the world with a thermo-nuclear warhead in less than 40 minutes [3]. Making a warhead from a ballistic missile hit a target on the other side of the world was no easy feat. To this day, doing so accurately is still a challenging problem. With shifting military strategy over the years came the preference for conventional weapons and as such, more emphasis on precision targeting. To improve accuracy, the capability to maneuver a reentry body became a critical requirement; hence, the birth of the guided reentry vehicle (RV).

The U.S. Air Force (the Army Air Corps until the official split in 1947) officially started its ICBM program in the early 1950s shortly after U.S. rocket propulsion made its debut with the successful launch of the Army Redstone Ballistic Missile in 1953 [4]. The success of this program was largely due to the technical leadership of Wernher von Braun and the "repackaged" technology of the German V-2 rocket [5]. At the same time, the Cold War crisis and the Soviet's victory of putting the first human in space fueled President John F. Kennedy's public decision in 1961 to support NASA's human spaceflight initiatives with the goal of sending a man to the moon via Project Apollo [6]. With this goal came the need to develop more advanced and effective technologies for reentry such that a vehicle carrying humans could safely return to earth. Breaking from the mold of the military's weaponized ballistic RV, NASA became the driving force behind most research and development (R&D) efforts in reentry technology with the Apollo reentry capsule. As the space race continued through the 1960s, rising costs associated with design, manufacturing, and operations of the Saturn V and other expendable launch vehicles started to have an economic impact on the government's budget. Therefore, the Space Shuttle program, starting shortly after President Nixon proposed the development of a reusable space transportation system

(STS) in 1972, officially kicked off with NASA's selection of Rockwell's STS division based out of Downey, CA to lead the design, development, and test and evaluation of the orbiter [7]. Although a magnificent engineering feat leading the way in reusable space access, the Space Shuttle turned out to be the most complex and expensive launch vehicle in the U.S. inventory [5].

Starting in 1994 under President Clinton's National Transportation Space Policy, NASA's Advanced Space Transportation Program (ASTP) [8] led a strong campaign to replace the aging and costly Space Shuttle. Numerous vehicle concepts were proposed that promised commercially viable space access by minimizing costs, and maximizing efficiency, effectiveness, and utility.

The concept of a fully reusable launch vehicle was one of the most controversial choices for future space transportation systems because of the many on-going challenges that faced industry. The most difficult challenges included: 1.) the process of integrating complex, state-of-the-art systems into a robust and cost efficient vehicle and 2.) financing breakthrough technological discoveries that would significantly minimize operational costs. Without adequate financial support from either Congress or private investors, materializing a fully reusable concept was out of reach. Although the Ansari X-Prize [9] showed some promise for a privatized space industry, investors are still wary of high risks associated with long-term investments in RLV development.

Today, the onset of a new race to the moon and back spawned NASA's initiative to develop a Crew Exploration Vehicle (CEV) [10] which has surprisingly reverted back to the pre-shuttle concept of the Apollo return module. Revitalizing space-plane concepts for a shuttle alternative, the CEV must satisfy requirements to serve as a manned, earth-orbit, moon, and mars return vehicle [11].

b. The Reentry Problem

The basic premise of the reentry problem is to autonomously and safely fly a vehicle from point A, the initial condition, to point B, the final condition, as depicted in Figure 1.1. This involves determining a flight path to take (i.e., trajectory optimization/generation), guidance system synthesis to follow the desired trajectory in the presence of disturbances and errors in the models used in the trajectory generation calculations, and maintenance of an attitude profile needed to follow that path (i.e., control).

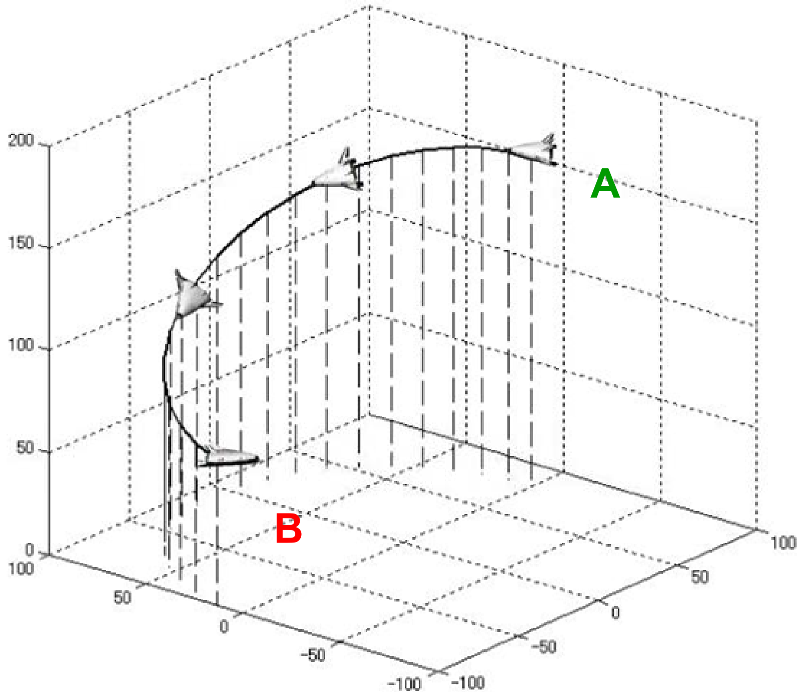


Figure 1.1 Reentry Problem: Autonomous Trajectory from Point A to Point B.

For an RLV that has entered space and needs to re-enter the earth's atmosphere, point A may be defined by conditions associated with a low-earth orbit. For example, the RLV may be de-orbiting from an altitude of approximately 170 km traveling at a speed of 27,870 km/h (7.74 km/s) in excess of Mach 25, roughly 8,149 km from its intended landing site, and will traverse approximately 20% the circumference of the earth from the deorbit point to landing. Since "reentry" implies entry into the earth's atmosphere, a more suitable initial condition for point A is the entry interface, defined by the vehicle's penetration into the sensible atmosphere. Since most of the atmosphere is below 100 km (62 mi), this "Karman Line" is the internationally accepted "demarcation" between the atmosphere and space [12], [13]. For testing feasibility of new G&C methods, it is even conceivable to start at sub-orbital altitudes. Although guiding a vehicle from point A to point B sounds like nothing more than a typical path planning and tracking problem, it must do so while keeping acceleration and heating loads within design limits. This is especially challenging due to its dynamic environment.

c. Specific Difficulties of Reentry G&C

In addition to the complex dynamics involved with the equations of motion for hypersonic maneuvering vehicles, there are various facets of reentry that make it a very challenging aerospace problem. Among these are path constraints due to environmental effects, state and control constraints based on vehicle limitations, and time constraints due to the nature of high-speed flight and online computational time requirements.

The primary cause of difficulties originates from the dynamic environment encountered when an RLV descends from space, through the entire atmosphere, and traverses large portions of the earth's surface. The vehicle encounters a large range of environmental conditions as it descends through the atmosphere, to include changes in pressure, temperature, density, and ionization. It also encounters gravity field variations, weather conditions (e.g.,. wind, precipitation, etc.), moving targets (the earth is spinning), etc. Among these aspects, the specific difficulties that pose the greatest challenge from a G&C perspective are related to vehicle limitations and uncertainties when maneuvering in such harsh and unpredictable environments. The high speeds associated with reentry can create high heat loads, g-loads, and pressure loads that ultimately must be tolerated by the vehicle's thermal protection system (TPS) and structure. Since RLVs in general are extremely limited in size, strength, and mass as a result of propulsion requirements for space access, the trajectory must be designed to minimize these loads, or at least not exceed operational limits. Originally, due to technology levels at the time of Shuttle design, the goal was to design a trajectory that minimized the loads so as to allow minimum weight for structure and TPS, thereby allowing more volume/weight for propulsion. For example, the Shuttle had a pre-defined angle-of-attack profile that required about 38 degrees during high heating rate and then down to 28 degrees to enhance crossrange performance [14]. Also related to limitations on weight, RLVs typically have a low lift-to-drag ratio (L/D) that reduces vehicle range and maneuverability. In terms of uncertainties, there are also numerous complicated flow-field interactions that affect the vehicle's aerodynamic characteristics. For example, rarified gas dynamics and ionization encountered in the upper atmosphere when traveling at high speeds is very difficult to predict, estimate, or measure. If developed correctly, the trajectory generator can accommodate these limitations by imposing path constraints and state constraints and simultaneously exploiting the physics of the problem by not imposing unnecessary constraints as discussed in Chapters II and III.

In addition to these flight path limitations and more consistent with typical aircraft, there are also electro-mechanical limitations inherent in the vehicle design. For example, the actuators have limited rates; the control surfaces have limited deflection ranges, etc. These types of limitations can be captured by imposing control constraints.

The final aspect that contributes to the challenge of the reentry problem is the concept of time. The only concern regarding flight time is the duration of time the vehicle spends in the atmosphere. Since the time-of-flight cannot be regulated by "thrusting" in the atmosphere for RLV reentry, time is dependent on the lift and drag characteristics of the vehicle. Too much time in the atmosphere at high-speeds can cause excessive heat loads beyond the vehicle's design limits. In this case, it is advantageous to bleed off speed as fast as possible and one way to do this is to increase drag. Increasing drag reduces L/D which ultimately reduces downrange distance; therefore, it is obvious that drag must be controlled to balance this tradeoff. Note that it is this regulation of drag that has formed the foundation of reentry G&C methods as discussed in the next section. Other flight time requirements may be from a strategic standpoint like turn-around-time. Other than sustained heat load time and strategic considerations, there are typically no requirements in terms of how long it takes an RLV to land; however, there is a requirement on computational time in the flight computer.

For a hypersonic vehicle traveling is excess of five times the speed of sound (Mach > 5), there is very little time for on-board computations. In terms of classical feedback control, this requires relatively high sampling rates. A demonstration of a new notion of feedback that doesn't require high sampling rates is part of the contribution of this dissertation and will be discussed later. A point of interest is that in the early 1960's, Wingrove determined from a survey of G&C methods that the guidance computational time required to maintain system stability must be on the order of two to tens of seconds [15]. In comparison, today's RLV systems, like the X-37 and X-40, require guidance loops to operate on the order of 10 Hz and control loops around 50 Hz. Potentially, time can affect the system's ability to calculate new guidance commands if new trajectories are desired; ability to accurately sense and/or estimate the vehicle's exact position (i.e., navigation); and ability to control the system. Overall, computational time constraints can affect the entire guidance, navigation, and control (GNC) system. Effectively and efficiently overcoming these various difficulties of the reentry problem has led to a large body of G&C R&D efforts.

2. G&C Methods for RLV Reentry

a. Traditional Approach: Based on Space Shuttle G&C

The ultimate goal of all G&C methods is to regulate aerodynamic forces and moments such that the constraints are not violated and that the vehicle safely reaches its final destination. Since numerous reentry G&C methods are modeled after the highly successful U.S. Space Shuttle entry guidance logic, it is first necessary to summarize the approach beginning with an overview of the various entry phases. This overview is primarily adapted from the NASA Shuttle reference manual [16].

(1) Shuttle Entry Phases. The Shuttle's normal trajectory guidance is separated into three primary phases: 1.) Entry, 2.) Terminal Area Energy Management (TAEM), and 3.) Approach and Landing (A/L). An altitude profile for these phases is illustrated in Figure 1.2.

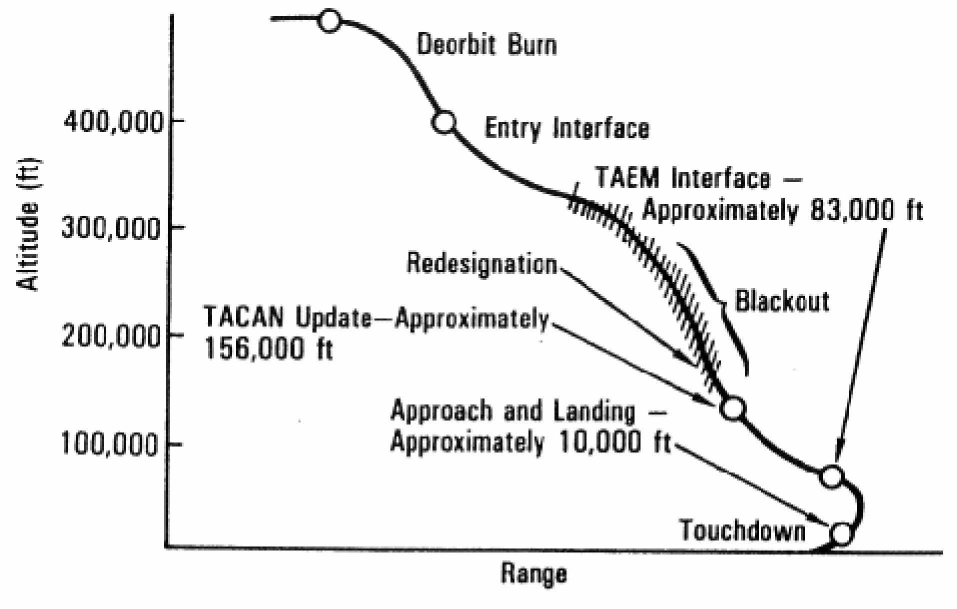


Figure 1.2 Space Shuttle Entry Flight Profile [16].

Beginning with its de-orbit at an altitude (*h*) of $557,000\,\text{ft}$, velocity (*V*) of $25,400\,\text{ft/s}$ and downrange distance (*DR*) from its intended landing site of $4400\,\text{nm}$, the Shuttle Orbiter enters the Earth's atmosphere at the defined entry interface $(h \approx 400,000\text{ ft})$ with a high angle-of-attack (~40 deg) to shield its body from the high heat loads. It continues its high-alpha descent through communications blackout, from approximately $265{,}000\text{ ft}$ to $162{,}000\text{ ft}$, until reaching the TAEM interface $(h \approx 85,000\,\text{ft}, V \approx 2,500\,\text{ft/s}, DR \approx 52\,\text{nm})$, where typical S-turn maneuvers (i.e., bank reversals) are performed to manage energy. The purpose of TAEM is to maneuver the Shuttle such that the proper conditions are achieved for runway alignment and to proceed with the A/L phase. By a series of bank commands, the Shuttle is able to adjust its drag acceleration while ensuring that structural and thermal limits are not exceeded. After the S-turns, the Shuttle follows a reference altitude profile and a series of waypoints to the Heading Alignment Cylinder/Cone (HAC) that serves to properly align the Shuttle's heading with the runway as shown in Figure 1.3.

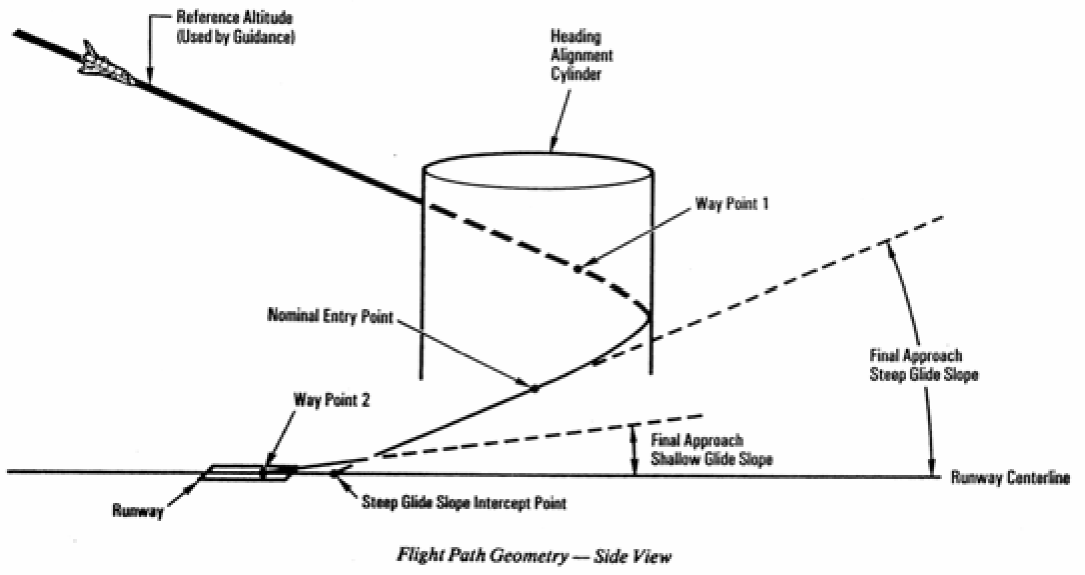


Figure 1.3 Space Shuttle Reference Altitude Tracking and Waypoint Guidance [16].

Depending on the approach to the runway ("straight-in" or "overhead") as shown in Figure 1.4, the Shuttle may use one of two HACs for alignment.

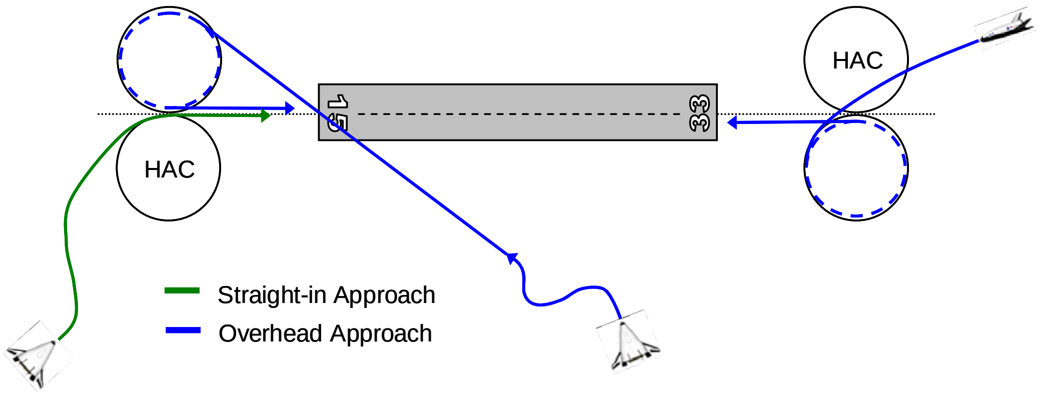


Figure 1.4 Typical Straight-in and Overhead HAC-Approaches for Shuttle.

The A/L interface $(h \approx 10,000\,\text{ft}, V \approx 290\,\text{ft/s}, DR \approx 5\,\text{nm}, CR \approx 1,000\,\text{ft})$ initiates the final approach sequence to landing. Except for the higher initial glide slope (-19 to -17 deg) at a higher airspeed and sink rate (~ 10,000 ft/min), the A/L phase consists of the characteristic final approach (i.e., shallow glide slope of ~ 1.5 deg), flare, touchdown, and rollout of a typical high performance aircraft. This "canned" A/L phase, shown in Figure 1.5, facilitates the use of an "autoland" sequence that initiates shortly before the TAEM-A/L switching point by the activation of "trajectory capture." Again, more details for each of these phases and their corresponding guidance segments can be found in Ref. [16].

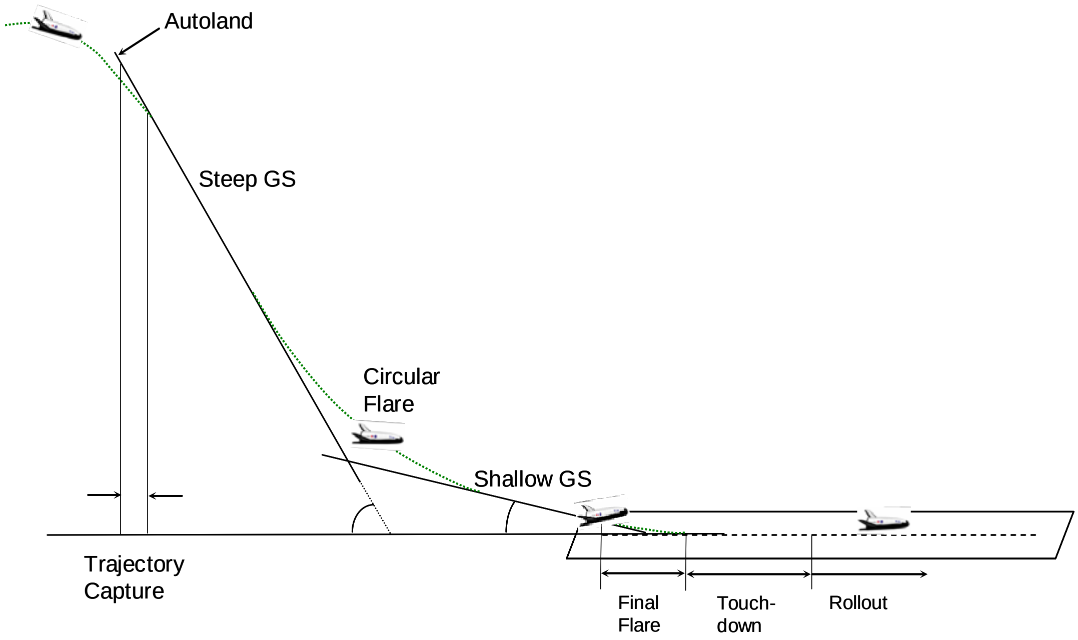


Figure 1.5 Approach and Landing Profile for "Autoland" System.

The relatively fixed geometry of the final approach segment facilitates the use of autoland guidance schemes. With a known geometric profile to a specific runway approach, the guidance scheme only needs to track the profile and achieve the desired touchdown conditions. For the Shuttle, the guidance system employs an "Autoland Shaping Processor" that predicts touchdown conditions to complete the desired tracking profile [17]. The guidance computer issues commands based on pre-determined waypoints such as altitude callouts in conjunction with sensor feedback from its inertial measurement unit (IMU) and the various navigation aids (air data system, tactical air navigation –TACAN, microwave scan beam landing system –MSBLS, and radar altimeter). Upon reaching 100 ft shortly after crossing the runway threshold to touchdown, on-board navigation uses radar altimeter measurements for guidance.

Stemming from earlier Shuttle autoland guidance [17]-[19], more recent efforts have made improvements in these schemes. For example, Barton and Tragesser developed the Autoland I-Load Program (ALIP) for the X-34 [20]. ALIP demonstrated that using initial and final dynamic pressure in a two-point boundary value problem reduced complexity and improved robustness that lends itself well as an onboard trajectory generation capability. Additionally, to handle off-nominal landing conditions such as large wind shear turbulence, Ha and Kim proposed an adaptive gain-scheduled proportional-integral-derivative (PID) control that is optimized using genetic algorithms [21]. Their fully automatic landing guidance algorithm which incorporates six-degree-of-freedom (6-DOF) dynamics can successfully land a simulated aircraft in bad weather conditions. Looye describes a series of controller functions that are defined based on stability, speed/flight path tracking, glide-slope guidance, and flare such that specified parameters can be determined via a multi-objective optimization problem [22]. Like the Shuttle A/L guidance scheme, Kluever has developed a trajectory-planning algorithm that computes reference flight profiles based on a series of geometric segments [23]. The Japanese (NASDA and NAL) went so far as to successfully demonstrate via Monte Carlo simulations and flight testing, an automatic landing algorithm in their Automatic Landing Flight Experiment (ALFLEX) that's incorporated into their HOPE-X RLV guidance system [24], [25] . Autoland routines are so common now that their methods, such as the use of a "glide-slope coupler" for guiding a vehicle on a predetermined flight path to landing [26], are found in elementary G&C textbooks. For this reason, the work presented in this dissertation does not address the landing phase, but rather prescribes a design and method that ensures final approach conditions are satisfied.

(2) Space Shuttle Drag-Based Guidance Logic. In taking less computer power than a modern digital wristwatch, the Shuttle entry guidance is a marvelous technical achievement wherein a drag-based tracking technique is used for guidance. The Shuttle's guidance logic, as described in one of the earliest public works by Harpold and Graves [14], is divided into five independent drag reference segments: two quadratic segments, a pseudo-equilibrium glide segment, a constant drag segment, and a linear segment. Each guidance segment is designed for a specific flight phase and then later linked together for the entire mission. For example, the two quadratic segments

are used for the region of aerodynamic heating that occurs at high speeds, the "temperature control phase" [14]. So, for each segment, a closed-form analytical expression is designed off-line for the desired drag acceleration profile that best satisfies the entry corridor and heating requirements. Then attitude commands (bank angle modulation) are generated to track the given profiles. The only flexibility in this guidance system comes from the capability to change the shape of each segment by changing the coefficients of the individual drag acceleration equations.

(3) Modified Drag-Acceleration Methods. Due to the relatively simplistic and successful drag-acceleration method used in Shuttle guidance, numerous efforts have pursued a similar approach. For example, Roenneke implemented a drag-verses-energy scheme whereby a linear feedback control law tracks a drag reference profile [27]. Likewise, Lu determined an optimal drag acceleration profile by parameterizing a reference trajectory and optimizing it as a piecewise linear function of energy [28],[29]. This piecewise approach forms simple analytical expressions that are tracked by a nonlinear controller. Follow-on efforts improved this work by developing a Receding Horizon Control (RHC) method, intended to have a simple implementation, that regulates a nonlinear system about a time-varying reference trajectory using closed-form approximations, again based on linearized time-varying (LTV) dynamics [30]. Later, Shen made this method faster by re-formulating the problem into two sequential, one-parameter search problems [31].

(4) Predictor-Corrector, Planner-Follower. Various efforts have pursued so called planner-follower architectures that combine predictor-corrector, originally proposed by Wingrove for abort control [15] with traditional tracking control (i.e., profile following). The idea is to use a trajectory generation scheme to plan a trajectory control sequence. This reference trajectory is then tracked by the follower. The predominant attitude control methods used for profile following consist of feedback linearization and dynamic inversion (DI) [32]-[34]. For example, Johnson and Calise have used feedback linearization with a neural network as a model reference adaptive controller for a trajectory following architecture called Pseudo-Control Hedging [35],[36]. Mease used feedback linearization of differential geometric control that allowed more global linearization of drag dynamics [37]. The main problem with these

approaches is their requirement for exact system knowledge (i.e., plant parameters). For example, dynamic inversion forms an augmented linear system by canceling nonlinear terms such that linear control techniques are applicable. With an imperfect model, substantial errors can result. For air vehicles where large uncertainties in aerodynamic forces and moments are inevitable, this method could potentially result in unstable, unrecoverable flight conditions. For this reason, other nonlinear feedback methods are being investigated. Some methods include various combinations of adaptive control, backstepping, and even more robust versions of dynamic inversion.

The culmination of the Advanced Guidance and Control (AG&C) program under direction of the NASA X-33 Project Office, Marshall Space Flight Center (MSFC), led to the comparison of various drag-based methods for the X-33. The methods consisted of baseline guidance (shuttle derived), Linear Quadratic Regulator (LQR), Predictor-Corrector (PC), Quasi-Equilibrium Glide (QEG) with PC, and an energy-drag technique called Evolved Acceleration Guidance Logic for Entry (EAGLE) [38]. Of these methods, EAGLE and QEG performed the best in NASA's high-fidelity Marshall Aerospace Vehicle Representation in C (MAVERIC) simulation environment.

b. Integrated Adaptive Guidance and Control Methods

With the realization of advances in computational power and numerical algorithms, recent research efforts for solving the reentry problem have focused on real-time, on-line trajectory generation, control reconfiguration, and guidance adaptation. For example, reconfigurable control based on an optimal control allocation algorithm has been performed at the Air Force Research Laboratory (AFRL) for vehicles such as the X-40, X-37, and X-33 [39]-[42]. Reconfiguration capabilities for RLV systems have also been rigorously pursued by such efforts as the AG&C Program and AFRL's Integrated Adaptive Guidance and Control (IAG&C) Program. More on these programs can be found in Refs. [38] and [42]. Schierman et al. went as far as combing a reconfigurable inner-loop with an adaptive outer-loop based on an online trajectory retargeting algorithm that makes use of polynomial neural networks (PNN) to store pre-computed trajectories that were solved via sequential quadratic programming (SQP). The online implementation was called Optimal-Path-to-Go (OPTG) guidance [43]. Other schemes for handling locked control surfaces involve a combination of traditional proportional-integral (PI) and quadratic programming-based control allocation [44],[45].

The drawback to using such methods again returns to the need of adjusting controller gains, enforcing trim, and tracking reference trajectories. In some cases involving failure scenarios, reconfigurable control and guidance adaptation may not be sufficient to recover full control; hence, trajectory reshaping is necessary to augment the guidance commands. It may also be required to abort the pre-planned mission altogether and retarget an alternate landing site. Work by Shaffer [47],[46] has integrated trajectory reshaping and retargeting with the reconfigurable control work of Oppenheimer and Doman [40] to demonstrate relatively fast computations of optimal trajectories under state-dependent control constraints resulting from control power limitations following control effector failures. Likewise, other approaches have begun to address the need for onboard trajectory reshaping [48]-[51].

Despite efforts at developing full on-line approaches, the concern for convergence and speed has motivated the use of off-line reference trajectories for tracking applications. Since the early days of space shuttle entry guidance, designers have been employing various reference trajectory tracking schemes as summarized in Sec. 2.a.(2)-(3). More recent efforts have employed optimal trajectory generation to solve for a reference input trajectory, then use other inner-loop control means to track the desired trajectory [52]-[54].

Another common approach for reentry trajectory tracking is the use of simple adaptive control theory for the design of model reference adaptive control (MRAC) techniques [55]. Although the popularity of adaptive control methods has risen significantly in the past five years, there are still various problems with their use. Wise provided a recent survey of adaptive control applications for flight vehicles that identified numerous open problems [56], summarized here for completeness:

1. Most MRAC schemes still need to schedule reference models or gain schedules for different flight phases since large changes in the flight envelope creates scheduling problems that can cause undesired dynamics;

2. Due to transient performance of adaptive systems, several parameter/matrices require tuning;

3. Trial-and-error methods associated with designing the adaptive system opens up potential problems with adaptation rates;

4. Various issues with sizing dead zones – for example, using fast learning rates addresses one problem while causing other problems with high frequency oscillations;

5. Filters used for structural mode suppression often reduces stability margins excessively;

6. Amount of gain and phase margins questioned as a result of robustness issues associated with feedback time delays.

In addition to adaptive methods, optimization methods, traditionally aimed at minimizing heat loads, has also become quite popular.

c. Trajectory Optimization Methods

Optimization techniques, such as dynamic programming, were used as early as 1961 to select nominal reference trajectories prior to reentry [15]. Bryson, providing perhaps the most influential bodies of work in flight trajectory optimization, implemented a steepest descent algorithm to design optimal lift/drag profiles that minimized heat with acceleration constraints [57]. Speyer combined perturbation theory with optimization to develop neighboring optimal solutions for the selection of nominal reference trajectories [58]. Hull and Speyer later implemented an augmented-Lagrangian method, a form of nonlinear programming (NLP), to compute maximum downrange and crossrange trajectories [59]. In addition to manually varying parameters to achieve a good guess in the form of a nominal trajectory, this method required the use of penalty functions to improve convergence properties and prevent weighting terms from shooting to infinity. These types of issues were highlighted by Pesch in a series of papers that culminated in an overview of off-line and on-line trajectory optimization for aerospace applications [60]. Stengel extended work involving energy-state approximations [61] by replacing velocity with kinetic energy and time with total energy to transform the reentry problem to one of fixed final energy and then solved using dynamic programming [62].

Energy-state approximations (ESA) neglect all state variables except energy so that only a single function needs to be minimized. This technique is embedded in NASA Ames Research Center's Hypersonic Aircraft Vehicle Optimization Code (HAVOC) that has been used to provide near-optimal trajectories [61],[63].

The most widely used trajectory optimization methods used today fall under the purview of either an indirect method or a direct method (see ChapterII., Sec. B.2). As thoroughly documented by Betts [64], each of these methods has their strengths and weaknesses. Perhaps the most common indirect method is the so called "shooting" technique. Simple and multiple shooting methods are based on transforming a two-point boundary value problem (TPBVP) into an initial value problem, varying the initial conditions, and integrating the differential equations (i.e., "shooting" them forward) until the endpoint conditions are satisfied. Numerous efforts have used shooting techniques to solve the reentry problem, including a modified simple shooting method that promises accurate and fast solutions [65]. Also, Zimmerman et al. separated a reentry trajectory into segments and applied two different guidance schemes (analytical heat rate and linear bank angle) that combined a classical shooting method with LQR tracking [66].

For direct methods, the direct transcription (i.e., collocation) became a popular choice for reentry work primarily due to robustness and flexibility as a result of not having to explicitly include necessary conditions.

Mease et al. used Boeing's Sparse Optimal Control Software (SOCS) to solve a reduced-order, optimal reentry trajectory planning problem [67] that led to the development of EAGLE [68],[69]. Since some trajectory optimization methods have stringent requirements for good initial guesses, elaborate techniques are being pursued to generate initialization trajectories. For example, Mooij has investigated parametric control variation analysis involving the use of statistical Taguchi methods to generate feasible trajectories [70]. Another common method used today is that of Model Predictive Control (MPC) and Receding Horizon Control (RHC) [71]. MPC and RHC are essentially the same techniques predicated on using a plant model to predict an output during a future time horizon and then compute the control commands by minimizing an objective function. Like DI-based methods, these methods also require very accurate plant dynamics as well as possible nonrobustness issues [72].

It has long been recognized that multiple, often natural, time-scales exist in certain nonlinear dynamic systems, particularly in aerospace problems. As such, a common technique for simplifying complex problems is to use time-scale separation to reduce the order of the dynamic equations. Based on parameterizing terms in the equations of motion by a scalar ratio of low and high-frequency modes, the idea of time-scale decomposition is to allow separate control design for the two modes and then piece them together for the full-order system. This technique has been employed on numerous flight trajectory optimization applications over the years [73]-[77].

To the best of the author's knowledge, the only attempt at solving the full 6-DOF reentry problem using collocation-based trajectory optimization also used a two-timescale decomposition approach to separate the fast and slow dynamics in order to capture the high frequency effects of the rotational equations [78],[79]. This technique appears to have been a successful implementation for generating optimal reentry trajectories, but consequently added another layer of guidance design and analysis. The proposed methodology of this dissertation is to use a PS method to solve the full 6-DOF reentry problem without relying on time-scale separation schemes to reduce the complexity of the problem, but instead offer a more simplified and direct approach.

d. New Approach

The proposed method departs from the traditional approach by abandoning the use of linearization, closed-form analytical approximations, reduced-order models, reference trajectory tracking, etc. and adopts a new approach based strictly on optimal control theory. It is now well established that many of the challenges on entry guidance fall under the purview of optimal control theory. Difficulties in solving optimal control problems have reflected difficulties in solving the entry problem.

The general idea of optimal control is to generate an optimum control history that drives the trajectory from an initial condition to some final state while minimizing (or maximizing) some performance index in terms of a cost function and subject to a series of constraints. The cost function is typically represented in a Bolza form,

$$J\left(\mathbf{x}(\cdot),\mathbf{u}(\cdot),t_0,t_f\right)=E\left(\mathbf{x}\left(t_0\right),\mathbf{x}\left(t_f\right),t_0,t_f\right)+\int_{t_0}^{t_f}F\left(\mathbf{x}\left(t\right),\mathbf{u}\left(t\right),t\right)dt \tag{1}$$

where the Mayer term, $E\left(\cdot\right)$, is considered a fixed cost and the Lagrange term, $F\left(\cdot\right)$, is a running cost integrated over time. This cost can then be minimized with respect to the states, **x**, the controls, **u**, and/or the clock times, t_0 and t_f, subject to constraint equations that includes nonlinear dynamics, $\dot{x}=f(x(t),u(t),t)$, paths constraints, endpoint conditions, and state and control limits. Additional details of this standard optimal control problem and its formulation are discussed in Chapter II and numerous textbooks [80]-[82]. With the reentry problem posed as an optimal control formulation, a nonlinear optimization tool is used to solve the problem. ASTOS, GESOP, POST, SOCS, OTIS, and DIDO are common trajectory optimization tools used for aerospace applications. In this work, the pseudospectral-based DIDO software package is employed. Additional details are discussed in Chapter II with appropriate references.

Overall, the goal of this approach is to transition to simple, autonomous G&C using real-time nonlinear trajectory optimization. This approach avoids G&C techniques that require labor intensive off-line design and analysis such as 1.) gain tuning and gain scheduling, and 2.) developing reference trajectories (i.e., nominal profiles) and selecting/designing control techniques to track them. Instead, the proposed approach significantly simplifies the development of the problem in the sense that it only requires an optimal control formulation that exploits the power of PS methods. More specifically, this approach avoids unnecessary model reductions, linearization, trajectory segmentation, and superfluous constraints such as equilibrium glide (i.e., trimmed flight) and zero sideslip. In addition, this approach, to some extent, dismisses the notion of separating the outer and inner-loop of the standard G&C system architecture by assuming an integrated, single-loop architecture as depicted in Figure 1.6. Chapter VI provides more details about these architectures.

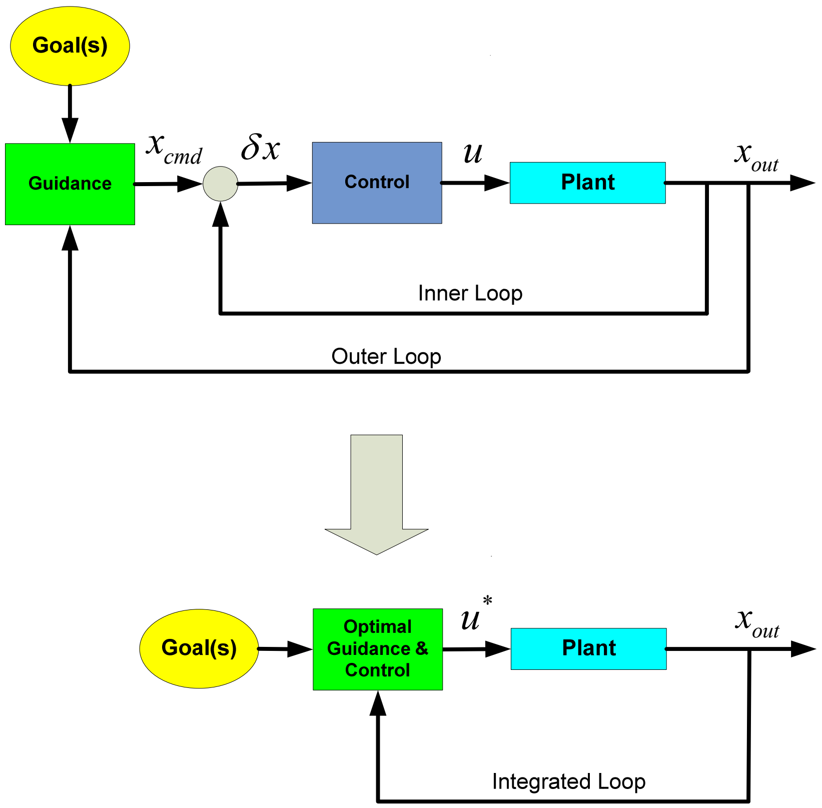


Figure 1.6 Typical "Separated" and New, "Integrated" G&C Architectures.

One immediate benefit of this proposed approach is that the integrated loop now assumes the role of the control allocation that is often separate from the trajectory optimization scheme in other inner-outer-loop architectures.

C. MOTIVATION

Despite recent efforts that have shown that reentry trajectory optimization problems are routinely solvable and viable for real-time, on-board applications [83]-[88], the perception that they are "not robust enough," [89] "computationally too complex to be solved in real time," "not meaningful," [90] and "ill suited for on-board implementation" [91] are still prevalent. Therefore, with such perceptions still prevalent in the literature, one of the primary motivations of this dissertation is to show that the proposed method is a viable approach for real-time, onboard applications and warrants further research investment.

The general motivation for considering optimality, in addition to the usual problems of constraints and nonlinearities, is that optimality dictates the engineering feasibility of a space mission. Whether or not it is explicitly stated, optimality is inherently required for feasibility, safety and other considerations. Traditional feedback control methods that are not based on optimality considerations may cause a system to perform short of its true capabilities. Fahroo has shown that traditional feedback control laws may also diminish safety margins when based on approximate, reduced-order models [85]. Another aspect of reentry safety is the size of the footprint: the larger the footprint, the safer the entry guidance as it implies the availability of additional landing sites for exigency operations. Thus, footprint maximization is part of the entry guidance requirements [86]. Consequently, entry guidance algorithms that are not based on optimal control compromise safety [87]. The following sections outline specific areas that spawned the motivation for the research work presented in this dissertation.

1. Enhanced Performance: Autonomy, Reliability/Safety, Adaptability, and Cost

A primary motivation of this dissertation is to address the endemic interest in "smart" systems that have more autonomous capability. As such, a goal of this research is to enhance performance by improving autonomy, reliability/safety, adaptability, and cost. Although there are various elements that contribute to making a system autonomous (planning, execution, system identification, adaptive control, sensor fusion, etc.), each element is influenced by three fundamental factors: mission complexity, environmental difficulty, and human interface. Regardless of the approach taken, each of these factors must be addressed. That is why the real "brain" of any autonomous system is the element of planning. In particular for vehicle motion, it is path planning (a.k.a. trajectory generation) that plays a critical role in the overall intelligence of the system. Planning is one of the primary elements of human intelligence that makes us extraordinary. In addition to our acute sensory perception, data processing, and motor control skills, most of our deliberate actions are a result of plans. Whether they are carefully and consciously or rapidly and subconsciously carried out, it is the intellectual capability of planning that allows us to accomplish a goal. Thus, for unmanned autonomous vehicles, path planning is not only required, but to effectively carry out certain missions requiring timeliness, real-time path planning is paramount.

Planning and scheduling is based on the ability of a system to generate low-level sequence of tasks that can accomplish formalized high-level goals. Traditional planning and scheduling has relied on humans to pre-program a set of actions to accomplish a pre-planned task or series of tasks. The focus for an intelligent system is to automatically determine the tasks required to accomplish overall mission objectives. Figure 1.7 illustrates various GNC methods that attempt to satisfy requirements and capabilities associated with intelligent/autonomous systems.

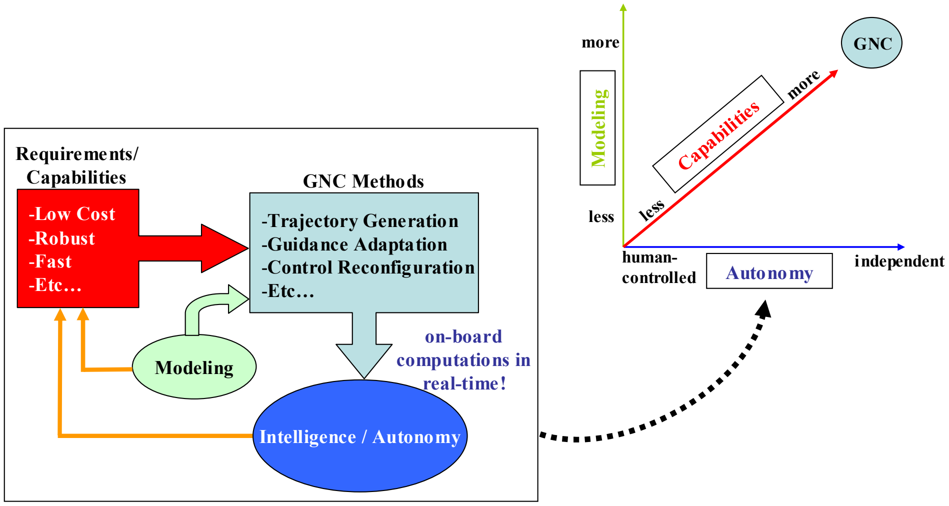


Figure 1.7 Using Guidance, Navigation, and Control Methods for Intelligence / Autonomy.

Note that modeling, in terms of both vehicle and environment, play a key role in how GNC methods provide enhanced capabilities. In other words, it doesn’t matter how autonomous the system is, if it operates based on a very crude model, then its effectiveness (i.e., capability) will be significantly hindered.

Additionally, another motivation for this research is to demonstrate how optimal control theory can be applied to an unmanned RLV in order to not only make it more intelligent, but to make its actions more desirable. Figure 1.8, adapted from the Joint Architecture for Unmanned Systems (JAUS) [92], shows the autonomy spectrum for various objectives of unmanned systems. Note the location of path planning is not very high and right on the chart. Historically and even today, a tremendous amount of man-hours go into planning trajectories. Since current vehicles have very limited on-board planning capability, much of the work must be performed manually off-line with a majority of work consisting of engineering judgment (i.e., human decisions). This is one reason why there is increasing demand to improve aerospace vehicle autonomy, to alleviate many man-hours of design and analysis. Preplanning is an acceptable approach for flight under nominal conditions with a perfect model; however, in the case of unforeseen failures, modeling errors, unpredicted uncertainties, large disturbances, etc., pre-planning alone is not sufficient. It is in these off-nominal, unplanned situations that real-time trajectory generation is paramount. Shifting to the far, top-right will not only make the system more autonomous, but can potentially replace other objectives like path following. It will be shown in later chapters that computing an optimal path in real-time provides benefits typically associated with feedback and adaptive methods; therefore, opening the possibility of replacing path following, adaptive behaviors, and disturbance rejection in the spectrum. Implementing an optimal path planning technique based on optimal control theory, can substantially improve the desired autonomous behavior portrayed in Figure 1.8. Note that this figure was originally developed for low-speed, powered unmanned aerial vehicles (UAV) where path planning is not as difficult as unpowered hypersonic entry.

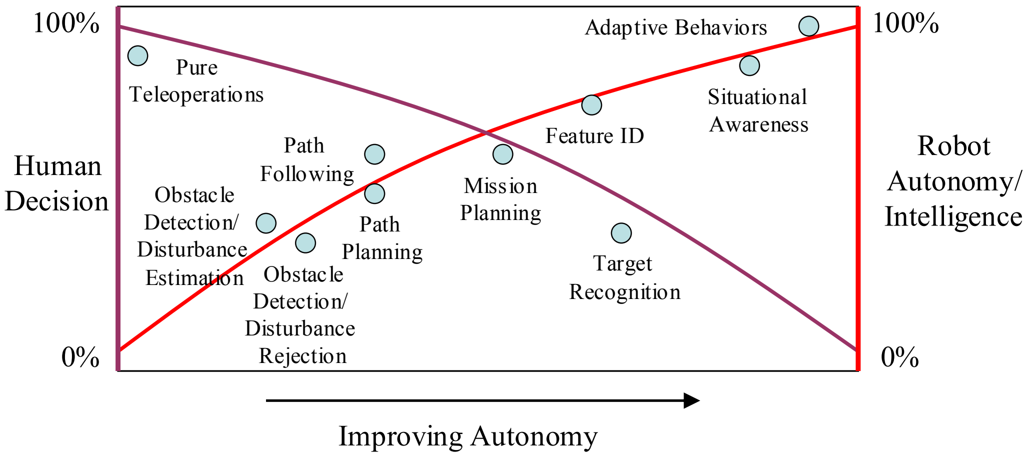


Figure 1.8 Autonomy Spectrum Adapted from Ref. [92].

Since optimal solutions are rarely intuitive, not to mention almost impossible to manually calculate, it is reasonable to assume that the capability of onboard optimal trajectory generation could potentially improve safety. For example, consider the February 1, 2003 Space Shuttle tragedy (STS-107) where the Orbiter failed during reentry over the western United States. Hypothetically, suppose that the astronauts and/or the NASA ground team knew of the damaged tiles shortly after their deorbit burn. At this point, the Shuttle is committed to reentry and there is nothing they can really do about changing their trajectory besides manual override. Even with manual control, they have only seconds to determine a new flight path and take a new course of action. If it was possible to save the vehicle by altering the flight path a certain way, maybe a way that would minimize heat load on the damaged area, then they or the ground team would never have enough time to calculate a new trajectory without the use of a real-time, optimal trajectory generator. Although this dissertation does not contain analysis to support this hypothesis, addressing this type of scenario with trajectory optimization methods is worth future investigation.

The key is to embrace autonomy and generality and avoid specific methods that have limited applications. With high costs associated with integrating complex and high risk systems, generality becomes a critical cost driver. By eliminating the need for vehicle-specific algorithms, the same guidance system should be usable in any vehicle in a "plug-and-play" fashion. This will reduce many aspects of cost – concept and prototype development, testing and verification, manufacturing and reproduction, maintenance, etc. This is a key advantage of the proposed method as compared to more recent research efforts involving the integration of different guidance methods to solve a complete launch vehicle problem [93].

2. Unified Approach: Simple, Effective, Robust, and Safe

A secondary motivation of this work is to provide a unified approach to RLV G&C that is simple, highly effective, robust, and safe. As illustrated in Figure 1.7, the development of new GNC methods is focused on achieving "smart" systems in order to provide more autonomous operations; however, in most cases, the methods are designed to a specific problem, and in some cases, to a specific vehicle. For example, methods that impose the so called equilibrium-glide condition [94], may be unnecessarily restricting maneuverability. Likewise, most reentry G&C methods force zero sideslip, especially in reduced-order models. Again, this limits maneuverability at lower speeds and lower altitudes unless a change to the algorithm is made mid-course to allow non-zero sideslip. This is one reason that many approaches to solving the reentry problem segment the trajectory and solve independent flight phases using different GNC methods for each phase. It is conceivable that the notion of stability/controllability is one of the main reasons why such methods hinge on forcing such conditions. In general, engineering assumptions, primarily an act of simplifying the problem, are usually problem-specific and vehicle-specific. By reducing the number and severity of assumptions, methods can be more general; hence, capable of solving various vehicle-independent problems. Also, numerous methods being pursued are very complex in the sense that they use very cumbersome mathematics, various combinations of analytical and numerical techniques, etc.

3. Modeling Issues in Footprint Generation

Over the last few decades, it has become customary to address the difficulties in entry guidance by way of reduced-order modeling and analysis [95],[96]. In a series of papers [85]-[87], Fahroo et al. showed that reduced-order modeling can compromise safety by predicting diminished footprints. That is, a reduced-order model may falsely imply that a preferred landing site was not reachable thereby generating a failure mode that could otherwise have been prevented. Thus, entry guidance algorithms that are not based on optimal control and reasonably high fidelity models compromise safety. This concept was further clarified by Shaffer et al. [47] who showed that optimal control techniques can also address trajectory reshaping in the presence of actuator failures. It has thus become increasingly clear in recent years that optimal trajectory generation is not a mere luxury but an absolute necessity for safe entry vehicle guidance. Furthermore, it is not sufficient to generate optimal trajectories for low-order models, but for sufficiently high fidelity vehicle and dynamical models. At the present time, it is unclear what qualifies as a sufficiently high fidelity model. For example, Shaffer et al. showed that high-fidelity aerodynamic models were crucial for ensuring that control failure induced state-dependent control constraints were represented for proper guidance in the Mach-alpha space. Fahroo et al. showed that high-fidelity dynamical models were crucial for footprint maximization. One way to address these issues is to construct models of increasing fidelity for a new kind of analysis that exploits all the modern tools available for solving optimal control problems. This is precisely the approach adopted in this dissertation.

Traditional methods avoid high-fidelity modeling based on reasons of “complexity” and computational speed, but as vehicles and computers become more advanced, higher-fidelity modeling offers more accurate solutions and these solutions become attainable with advances in numerical methods and computational power. Reduced order models typically produce adequate results but are providing near-exact solutions to approximate problems. It may be more advantageous to provide an approximate solution to an exact problem. Yet, it is often the case that the reduced-order models are providing wrong answers. This may also be true for complex aerospace vehicles where the rotational dynamics and translational kinematics operate at different frequencies. Typically, as long as the frequency separation is adequate, the reduced order models work fine. The problem comes from changes in the forces due to failures and rotational tracking requirements as well as new constraints that may arise from failure conditions. Regardless, various research efforts have shown that inaccurate and/or misleading solutions can result from using low fidelity modeling [85],[26],[97]. For example, for the specific application of a RV in atmospheric flight, some rotational effects may be absent or conflicting when considering inner/outer-loop simulations. Current, more powerful, numerical methods can efficiently provide accurate solutions to large systems of nonlinear Ordinary Differential Equations (ODEs).

Figure 1.9 illustrates an example of potential problems with using low fidelity models for footprint generation. The two regions show “exaggerated” maneuverability envelopes for a low-fidelity model and a high-fidelity model. Here, the high fidelity model expands the envelope (reachable set) which is clearly an increase in performance whereas the low-fidelity model simplifies the physics of the problem and consequently reduces the maneuverability envelope. Not only could the envelope be smaller, but it could incorrectly portray what is physically possible as indicated by the region around Air Field 3. Starting from the initial condition (IC), Air Field 1 is reachable under both envelopes, but Air Field 2 and Air Field 3 are not.

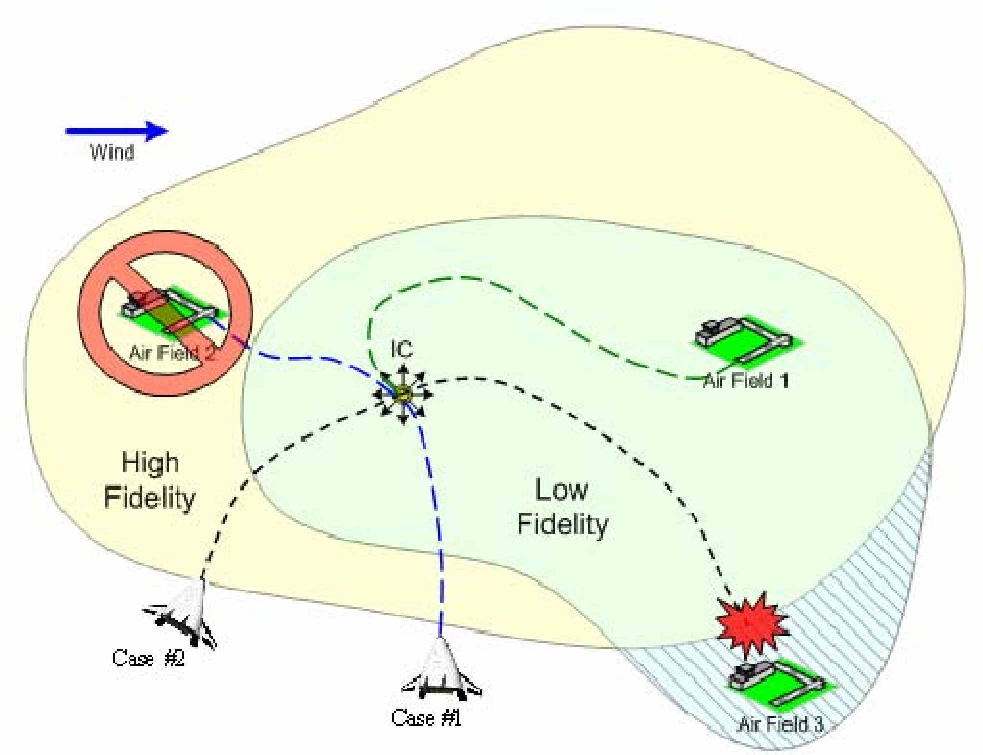


Figure 1.9 Maneuverability Envelopes Illustrate Problems with Low-Fidelity Solution.

Consider two specific examples indicated by "Case #1" and "Case #2." First, assume the vehicle is originally planning to land at Air Field 2 and experiences a failure corresponding to the "IC" mark. For the low-fidelity model, it is no longer viable to reach Air Field 2, but it can reach the alternate, Air Field 1. In this case, even though it *falsely* thinks it cannot reach Air Field 2, it still may have a feasible alternative. In comparison, the high-fidelity model can reach both. Second, assume the vehicle is originally heading to Air Field 3 and experiences a failure corresponding to the "IC" mark. Now, the low-fidelity model provides incorrect information, a *FALSE POSITIVE* that it can reach Air Field 3 when in fact, it cannot, so it will unsuccessfully attempt to make it. This case will end in, the very least, an emergency situation that could have been prevented simply by using the high-fidelity model. This is important now more than ever since current research is developing on-board, replanning and retargeting guidance schemes. Even if using low-fidelity-based methods, the vehicle still needs high-fidelity answers during real missions to definitively determine feasibility.

4. Problems Associated with Decoupling Inner/Outer Loops –The Case for “Integrated” G&C

Traditionally, inner and outer loops are decoupled such that they are actually being solved separately. This may be fine for simple linear systems with sufficient frequency separation, but for complicated nonlinear systems there may be more to the inner-outer-loop interplay that can have detrimental effects on system performance. As mentioned previously, Shaffer’s work revealed a disconnect between the inner-loop control allocator and outer-loop guidance when solving an X-33 RLV reentry problem for footprint determination under nominal and control effector failure scenarios [46]. This work solved a 3-DOF X-33 trajectory optimization problem using interpolated aerodynamic data that incorporated 6-DOF wing, body and trim effects from a given vehicle flight condition and an optimized effector displacement vector. This essentially decoupled the outer loop from the inner loop, but still provided the outer loop with a feasible range of Mach and angle-of-attack combinations along with the 6-DOF aerodynamic effects. However, conflicting cost functions between the inner-loop optimal control allocation problem and the outer-loop optimal guidance problem caused the failed vehicle to provide better performance than the nominal vehicle. This specific case was a result of improperly defining the control allocator cost function to minimize control surface deflections; however, there could be additional underlying sources of error. For example, the inner-loop control algorithm uses a piecewise linear programming (PLP) method to approximate the nonlinear optimal control allocation whereas the outer-loop guidance uses a direct method to solve the full, nonlinear system.

Additionally, since many efforts associated with the development of reentry G&C add a requirement for trimmed flight; the next section summarizes steady-state flight and trim conditions and addresses why trim may be unnecessary when utilizing optimal control to solve the full 6-DOF problem.

5. Requirements for Trimmed Flight

a. Steady-State Flight and Trim Conditions

In the dynamical sense, steady-state implies that there are no changes over time and as such the forces and moments acting on the vehicle’s body must be constant or zero. Likewise, the angular rates (p,q,r) and the aerodynamic angles must be constant and hence their derivatives are zero. Steady-state "trimmed" flight is an equilibrium condition based on the balance of these forces and moments such that quasi-static conditions are achieved. For example, pressures and shear stress distributions over the wing-body create a longitudinal pitching moment that act on the vehicle. Trim is attained when this pitching moment is cancelled as a result of forces generated from control surface deflections that counter the wing-body moment and achieves static equilibrium in the pitch axis. In the conventional sense, longitudinal static stability is achieved when the pitching moment at zero lift is positive $(C_{m_0} > 0)$ and the corresponding slope of $C_m \; vs. \, \alpha$ is negative $(\partial C_m / \partial \alpha < 0)$. Note that a control surface can also be used to move the vehicle to a different, pre-determined equilibrium point. Also, the equilibrium point is typically influenced by wind-relative incidence angles such as the angle-of-attack(α). For example, the angle-of-attack that causes zero pitching moment is termed the "trimmed" angle-of-attack(α_e). More on this can be found in Refs. [98]-[100].

Although static stability is not sufficient to ensure dynamic stability, a dynamically stable vehicle must always be statically stable [100]. For example, when the vehicle initially tends towards its equilibrium position after a disturbance (statically stable) and continues over time to the equilibrium, then it is dynamically stable. However, if after its initial turn towards the equilibrium point it diverges away, then the vehicle is dynamically unstable yet statically stable.

b. Maneuvering Flight

For maneuvering flight, control surface deflections create non-equilibrium accelerated motions (rotational and translational accelerations). In this case, there is not necessarily a balance of forces and moments unless a specific maneuver is perfectly coordinated such that the state derivatives are zero. For example, this occurs during a "steady-state, coordinated turn" discussed later.

Control methods employing linearization techniques often enforce steady-state "trimmed" flight when studying aircraft dynamics and performance since they consist of linearizing about some nominal equilibrium set-point. Also, flight simulations often prefer trimmed initial conditions. Historically, the design of flight control systems was also based on trimmed flight conditions. For a specified flight envelope, various operating conditions would be set as nominal points at which the equations of motion are linearized. The linearized equations corresponding to these set-points would then be used to design a controller with appropriate gains to form a set of “scheduled” gains that span the entire flight envelope or at least the effective region that the control is active.

c. Equilibrium Glide and the Reentry Problem

For typical reentry trajectories, a large portion of the trajectory is considered to be an “equilibrium glide” whereby the vertical component of acceleration is small along with a small and slowly varying flight path angle. Essentially, glides become shallower as the L/D ratio increases with the maximum glide being associated with the smallest equilibrium glide angle that occurs at maximum L/D. This is approximated in the 3-DOF dynamic equations of motion by setting $\dot{\gamma} = 0$, referred to as the equilibrium glide condition (EQC) [31]. As such, a simplified 3-DOF equation for $\dot{\gamma}$ reduces to an algebraic equation in terms of altitude, velocity, and bank angle that allows the velocity-altitude profile to be shaped by the corresponding bank angle. Also, as pointed out in Ref. [94], the flight path angle and bank angle still vary with time during this so called equilibrium glide and is therefore re-termed “quasi-equilibrium glide condition (QEQC).” A formal analysis of these conditions can be found in reference [94].

d. Requirements for Coordinated Turns

Coordinated turns, or coordinated maneuvers in general, are traditionally preferred for manned flights in order to provide an acceptable comfort level [26]. Also, maximizing aerodynamic efficiency and minimizing structural loads is a performance requirement that is usually achieved by minimizing the sideslip angle throughout the flight. A perfectly coordinated turn is one in which the sideslip angle is zero as a result of the side force from gravity being counteracted by the vehicle’s yawing-motion. Also, a steady-state turn can be reached after the transient accelerations subside such that the first and second state-derivatives are zero.

Although structural loads are still a concern for unmanned vehicles, comfort is not. This allows more flexibility in the maneuvers such as faster rates and slightly larger loads. An excellent example of the potential performance benefits of unmanned versus manned flight is demonstrated by the results of a fuel-optimal trajectory simulation for UAVs [101]. This paper shows that the optimal endurance path is an up and down oscillatory motion as a result of full-thrust, idle-thrust “bang, bang” control.

Although sideslip is not usually desired at high velocities such for the case of reentry, forcing this condition in any simulation, especially one seeking an “all-or-nothing” objective such as making a landing field, places unnecessary limitations on the obtainable flight path. From a practical flight perspective, there are times when purposely reducing aerodynamic efficiency (e.g., by inducing sideslip angle) is desired. For example, in a higher than normal approach to landing, a “slip” can be used to quickly reduce altitude in order to not overshoot the intended landing spot. Therefore, forcing a zero sideslip condition in the model could result in the false notion that a landing site or final target condition is not reachable when it may be that only a small sideslip is needed. This relates to the whole idea of why using a reduced-order model may not be solving the correct problem or inducing unnecessary constraints. Besides, modeling the appropriate structural limits as constraints is a more practical approach that does not limit the “realism” of the model by use of “artificial” constraints such as zero sideslip.

e. Nonlinear Control and Issues with Trim Requirement

Current and future efforts will likely make use of nonlinear control methods that steer away from linearization and gain scheduling, especially for autonomous vehicles that operate over a large range of flight conditions that have strong nonlinear behavior.

For control of complete nonlinear systems, such as the reentry problem using the full 6-DOF nonlinear equations of motion, enforcing trim conditions is not necessary. Traditionally, flight control algorithms were designed around trim conditions at various points in the operational flight envelope primarily because linearization techniques were employed. In addition, an advanced nonlinear simulation should be able to not only start at arbitrary initial conditions, but also to find equilibrium points from which to start.

In the case of control surface failures, it may be required to first stabilize the vehicle by imposing trim conditions via a control allocation scheme. Such schemes usually attempt to minimize some cost function with the goal of finding the control that produces the desired trim condition with control inputs that are as close as possible to a set of desired control inputs. Although this condition will vary with flight condition, in most cases it involves ensuring that certain state derivatives are zero (i.e., steady-state). Once equilibrium is achieved, maneuvering may be possible assuming there is enough control authority available with the given failure. The challenge then becomes flying with degraded performance and possibly limited characterization/understanding of the failure and its effect on autonomous system performance.

f. Reconfigurable Control via Control Allocation

Recent efforts have focused on reconfigurable control through use of advanced adaptive control and control allocation methods. For a vehicle that encounters a control failure, such as a locked control surface (i.e., fixed deflection position), the goal of the on-board flight control system is to reconfigure the remaining controls such that the vehicle is still controllable and can still track the desired reference inputs provided by the guidance computer. Hence, the control allocation scheme is responsible for determining how to distribute or to "allocate" the remaining control effectors such that the desired moments or accelerations are generated. This problem is only valid for an overdetermined system (i.e., the number of control effectors is greater than the number of axes to be controlled). For example, given a pre-failure and post-failure nonlinear control system, $\dot{x} = f(x,u,t)$, represented as

$$\boxed{\begin{array}{c}\dot{x}^* = f(x,t) + g(x,t)u^* \\ \text{pre-failure}\end{array}} \longrightarrow \boxed{\begin{array}{c}\dot{x} = f(x,t) + g_r(x,t)u_r + g_d(x,t)u_d \\ \text{post-failure}\end{array}}$$

where the * indicates the pre-failure state, control, u_r is the remaining control, and u_d is the postfailure influence represented as a disturbance input. Then for the pre and post-failure system to provide the same performance, the allocator should find u_r such that $g_r(x,t)u_r + g_d(x,t)u_d = g(x,t)u^*$. Note that for an "integrated" G&C system capable of real-time trajectory generation, tracking a reference input is not required since the reference is continuously updated…similar to feedback in the guidance loop of a traditional inner-outer-loop G&C configuration.

g. Trim Requirements

It is usually desired to stabilize the vehicle immediately after a control failure (e.g.,, jammed effector) by driving the system to some equilibrium point. Using a control allocator to achieve this trim condition provides knowledge of the effector positions and hence allows calculation of their contributions to forces which heavily influence the performance capabilities of the vehicle. For example, assuming steady-state flight with coordinated-turns (zero sideslip) as the desired trim condition, the control allocator will try to zero out the roll and yaw moments and balance the pitch moment produced by the adverse effects of the failure. Balancing the moments result in zero angular accelerations in pitch, roll, and yaw. However, trim conditions are typically used if the objective is to track a reference trajectory or the failure causes significant instabilities (depends on flight configuration at time of failure) that require stabilizing about some known equilibrium state. Otherwise, in the case of maneuvering flight, forcing trim and in this case, forcing zero sideslip, may limit the maneuverability envelope of the vehicle.

The key is knowing where the effectors are positioned and in a practical sense, this can be done by smart actuators that are available today. For a given failure and known effector position, the forces and moments can still be approximated assuming the surface is locked in a position that lies in the range for which aerodynamic data exists (i.e., within control surface limits). Effectors can lock or saturate, but as long as axis saturation does not occur, the vehicle will remain controllable. It may be desired to at least check that trim is attainable in order to stabilize the vehicle to some known steady-state equilibrium condition, but if the full 6-DOF dynamics are considered in a trajectory optimization problem (based on optimal control and NLP), the allocation of controls to achieve the objective would be based solely on the control constraints. In this case, the effector failures are mapped to changes in the control constraints and the optimization problem will continue to minimize the objective function with a modified set of constraints. This concept is illustrated by the “Re-Model Constraints” block of Figure 1.10.

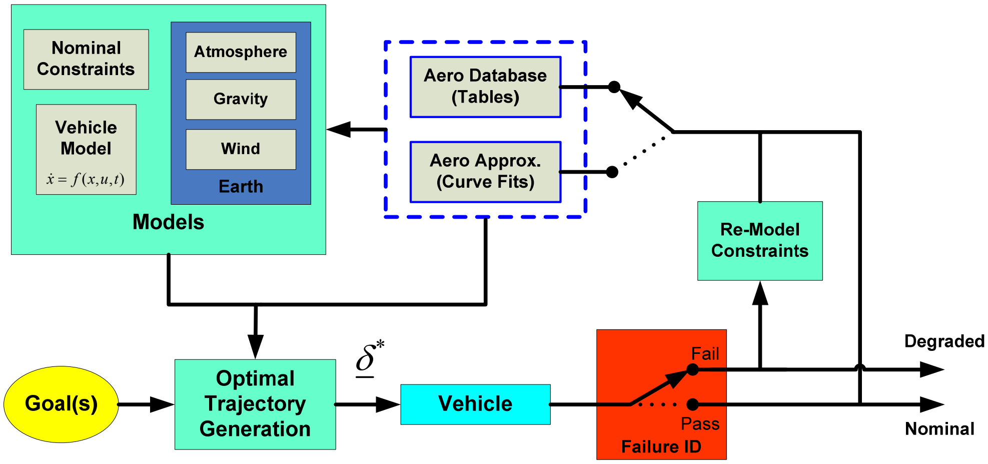


Figure 1.10 Conceptual Block Diagram of Optimal Trajectory Generation for Handling Control Failures without Control Allocation.

If there is a problem with convergence as a result of vehicle instabilities or erratic maneuvers, it is possible to reduce angular velocities by adding a penalty to the optimization objective function in the form of a weighted Lagrange cost. However, if the optimization problem formulation includes realistic constraints, then the vehicle should not perform any maneuvers that are not realizable by the control system.

D. CONTRIBUTIONS

The primary contribution of this dissertation is the development of a new, unified approach for the autonomous G&C of the highly constrained and nonlinear RLV reentry problem. This work combines many of the emerging concepts in real-time optimal control to address the difficulties in entry guidance. To this end, this work circumvents the traditional problems that arise in entry segmentation (such as the concept of TAEM), reference trajectory tracking, and separate G&C-loop architectures and instead embarks on using a single algorithm to guide the vehicle from its entry conditions all the way to a neighborhood of the landing site. This vehicle-independent approach relies on a PS method wrapped into a user-friendly software package that offers simplicity, generality, and robustness.

Second, the proposed method is based solely on optimal control theory and entails the development of a high-fidelity, 6-DOF model that addresses an important issue in model fidelity and its impact on safety. This 6-DOF approach is unique because it replaces or augments the traditional, separated inner and outer loop G&C architecture with an integrated, single-loop structure for generating optimal solutions. As an alternate approach to the full, 6-DOF architecture, this research also provides a means of integrating 3-DOF optimal guidance with 6-DOF tracking control.

Third, part of this research involved expanding the capabilities of trajectory optimization and as such, this dissertation demonstrates the viability of using the PS-guidance method for onboard applications to include footprint generation, intelligent/autonomous trajectory generation to a designated landing site, and retargeting alternate landing sites as a contingency operation.

In addition to some of these onboard applications, a unique contribution of this research is the implementation of a PS-feedback method to the RLV reentry problem. This real-time implementation demonstrates the method's viability for use in optimal guidance algorithms that require corrective maneuvers from the perturbed trajectory in addition to validating new theoretical developments in the field of PS-based optimal control.

E. DISSERTATION ORGANIZATION

The organization of this dissertation attempts to lead the reader through a logical progression of developments culminating with the implications of the high-fidelity, trajectory optimization capabilities presented throughout the chapters. **Chapter II** begins with summarizing the theoretical foundations and highlighting some mathematical preliminaries essential in understanding the framework for pseudospectral optimal G&C and how it is used to solve the RLV reentry problem. **Chapter III** then introduces the RLV model used for this study, a succinct review of the development of the standard 6-DOF equations of motion, their limitations in the context of optimal control problem formulation and optimization suitability, and proposes an alternate dynamical model specifically for flight vehicle trajectory optimization. Chapter III also includes brief discussions on modeling the environment, model reduction from 6-DOF to 3-DOF, and a homotopy path to modeling including emphasis on model fidelity and its implications on solution accuracy. **Chapter IV** presents results from a 3-DOF RLV model with the intent of testing the various G&C ideas for possible 6-DOF implementation (i.e., proof-of-concept). This includes open-loop, optimal trajectory optimization that demonstrates intelligent path planning, on-board retargeting capability, and closed-loop, optimal trajectory optimization that demonstrates the efficacy of a PS-feedback method as well as validates the conjecture that real-time open-loop is closed-loop. **Chapter V** serves as "transitioning" work from a 3-DOF to a 6-DOF optimal G&C architecture by demonstrating optimal trajectory tracking. **Chapter VI**, the crux of this research, presents the 6-DOF results for various model formulations and exposes some important issues involved with high-fidelity trajectory optimization for RLVs. For this chapter, particular emphasis is placed on feasibility analysis along with numerical considerations. **Chapter VII** concludes with a summary of this research work and the original contributions provided. **Chapter VIII** ends the dissertation with some brief discussions on various open issues and opportunities for future work. Chapter VII also includes a list of extended applications of the ideas and methods presented herein.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THEORETICAL FRAMEWORK FOR OPTIMAL GUIDANCE AND CONTROL

A. OVERVIEW

Since the majority of this research work depends on the application of nonlinear optimal control theory, it is necessary to summarize the details of the approach and highlight some of its benefits over more traditional methods. Traditional methods involve finding closed-form solutions to Hamilton-Jacobi equations or the canonical Hamiltonian equations. With the advent of more powerful computers, computational methods are now more widely accepted as a direct result of the numerous difficulties associated with solving a state and control-constrained nonlinear optimal control problem (NL OCP). Numerical methods for solving the NL OCP typically fall into the categories of indirect or direct methods with many variations of each developed over the years (see Sec. B.II). Due to its exponential convergence rate and successful history of solving complex NL OCPs, a general-purpose method, the Legendre pseudospectral (PS) method, is the method of choice for solving the class of reentry problems presented in this work.

To solve the NL OCP, a spectral algorithm known as the Legendre PS method is employed in a MATLAB-based software package called DIDO [102] that was developed by Fahroo and Ross of the Naval Postgraduate School. This method discretizes the problem and approximates the states, co-states and control variables by use of Lagrange interpolating polynomials where the unknown coefficient values coincide with the Legendre-Gauss-Lobatto (LGL) node points. After this approximation step, a nonlinear programming (NLP) solver (SNOPT) [103] based on sequential quadratic programming (SQP), solves a sequence of finite-dimensional optimization problems that capture the full nonlinearities of the system. The spectral algorithm uses the NLP solver sequentially to solve the OCP efficiently [104],[105]. Unlike other traditional methods, the PS method uses discretization to transform large-scale optimization problems into significantly smaller-scale problems; hence, improving both speed and convergence properties.

Before outlining the specific PS solution methodology, it is first necessary to define some mathematical preliminaries regarding optimal control theory such as: 1.) the

NL OCP, 2.) indirect and direct methods, 3.) necessary optimality conditions and the minimum principle, 4.) the Karush-Kuhn-Tucker (KKT) Theorem, 5.) the NLP problem and 6.) verification of feasibility and optimality.

B. OPTIMAL CONTROL THEORY

Optimal control theory, sometimes referred to as dynamic optimization [106]-[109], grew out of the Calculus of Variations (COV), perhaps one of the most influential mathematics of optimization theory. Like COV that deals with extremal functions, the basic premise of optimal control involves finding the control histories that minimize a performance index subject to constraints in the form of differential equations. It is the intent of this section to generalize the NL OCP, define the necessary optimality conditions, and state the fundamental principles that form the framework for the solution methodology implemented in this work.

1. The Nonlinear Optimal Control Problem (NL OCP)

As with any dynamical optimization problem, the cost function (a.k.a. objective function), dynamic constraints (i.e., governing equations of motions), mixed state-control path constraints, endpoint constraints (i.e., boundary limits on initial/final conditions), and any additional equality or inequality constraints (on states and/or controls) must be defined. As such, the general OCP for trajectory generation is fully posed with the respective constraints in the following manner:

$$\min_u J\left(\underline{x}(\cdot),\underline{u}(\cdot),\tau_0,\tau_f\right)=E\left(\underline{x}\left(\tau_0\right),\underline{x}\left(\tau_f\right),\tau_0,\tau_f\right)+\int_{\tau_0}^{\tau_f} F\left(\underline{x}\left(\tau\right),\underline{u}\left(\tau\right),\tau\right)d\tau$$

$$\text{subject to}\begin{cases}\dot{\underline{x}}(\tau)=\underline{f}\left(\underline{x}(\tau),\underline{u}(\tau),\tau\right)\\ \underline{h}^L\le\underline{h}\left(\underline{x}(\tau),\underline{u}(\tau),\tau\right)\le\underline{h}^U\\ \underline{e}^L\le\underline{e}\left(\underline{x}\left(\tau_0\right),\underline{x}\left(\tau_f\right),\tau_0,\tau_f\right)\le\underline{e}^U\\ \underline{x}^L\le\underline{x}\left(\tau\right)\le\underline{x}^U\\ \underline{u}^L\le\underline{u}\left(\tau\right)\le\underline{u}^U\end{cases}\tag{2.1}$$

The goal is to find a state-control function pair, $\left\{x\left(\cdot\right),\, u\left(\cdot\right)\right\}$, or sometimes clock times, τ_0 and τ_f, that minimizes the performance index represented by the Bolza form, $J\left(\cdot\right)$, consisting of either a Mayer term, $E\left(\cdot\right)$, a Lagrange term, $F\left(\cdot\right)$, or both as stated in

Eq. (2.1). The general OCP dimensions can be expressed in terms of the number of state and control variables, N_x and N_u, respectively, such that $\underline{x} \in \mathbb{R}^{N_x}$ and $\underline{u} \in \mathbb{R}^{N_u}$. Also, it is assumed that the following functions are continuously differentiable with respect to their arguments.

$$\begin{aligned}
E &: \ \mathbb{R}^{N_x} \times \mathbb{R}^{N_x} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \\
F &: \ \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \rightarrow \mathbb{R} \\
f &: \ \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \rightarrow \mathbb{R}^{N_x} \\
h &: \ \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \rightarrow \mathbb{R}^{N_h} \\
e &: \ \mathbb{R}^{N_x} \times \mathbb{R} \rightarrow \mathbb{R}^{N_e}
\end{aligned} \tag{2.2}$$

After formulating a problem according to Eq. (2.1), it can be solved by either indirect or direct methods.

2. Indirect and Direct Methods

Numerical Methods for solving the NL OCP typically fall into one of two categories: indirect or direct methods [110]. Indirect methods are based on root-finding techniques associated with numerical search of finding boundary values for states and costates that minimize cost and satisfy constraints. Such root-finding techniques are the common Newton's Method that must satisfy necessary conditions like $\nabla f(x) = \frac{\partial f(x)}{\partial x} = 0$. On the other hand, direct methods attempt to minimize some objective function that results in a sequence of improving solutions until attaining an optimal value. In terms of optimal control theory, the indirect method requires the COV to establish and solve the necessary optimality conditions. Depending on the complexity of the problem, this can often be a challenging task. Despite some successful implementations of indirect methods [110]-[114], they still have numerous problems. One of the main drawbacks (mostly due to practical limitations) with indirect methods is that they require relatively accurate initial guesses in order to provide feasible trajectories [115],[116]. Unlike the direct method, the indirect method also requires the explicit derivation of necessary conditions, to include the adjoint, control, and transversality as presented in a general context below.

3. Necessary Optimality Conditions and the Minimum Principle

The necessary conditions needed for optimality first requires the formulation of the *control Hamiltonian*

$$H \equiv H(\lambda(\tau), x(\tau), u(\tau), \tau) = F(x(\tau), u(\tau), \tau) + \lambda(\tau)^T f(x(\tau), u(\tau), \tau) \qquad (2.3)$$

where $H : \mathbb{R}^{N_x} \times \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times \mathbb{R} \to \mathbb{R}$, $F(\cdot)$ and $f(\cdot)$ correspond to the OCP formulation in Eqs. (2.1) and (2.2) as the Lagrange cost and the vector field for the right hand side of the differential equations of motion, respectively, and the states and costates given as $\underline{x} \in \mathbb{R}^{N_x}$ and $\underline{\lambda}^T \in \mathbb{R}^{N_\lambda}$, respectively. The Hamiltonian adjoins the state dynamics to the Lagrange term in the cost function using the Lagrange multiplier, λ. Lagrange multipliers are the duals of the primal solution and represent sensitivity to the constraints – "tightness" of constraints [117]. For example, if a multiplier is large, then it indicates that the corresponding constraint is important for the solution.

When minimizing the cost function to determine the optimal states, costates, and controls (x^*, λ^*, u^*), the following conditions are required for optimality. First, the partial derivative of the Hamiltonian with respect to the Lagrange multiplier must recover the state dynamics given by the *state equation*

$$\begin{aligned} \dot{\underline{x}}(\tau) &= \frac{\partial H(\underline{\lambda}^*(\tau), \underline{x}^*(\tau), \underline{u}^*(\tau), \tau)}{\partial \underline{\lambda}(\tau)} \\ &\Rightarrow \textit{RHS of dynamic equations} = \underline{f}\left(\underline{x}(\tau), \underline{u}(\tau), \tau\right) \end{aligned} \qquad (2.4)$$

Next, the costates must "shadow" the state dynamics and satisfy the *adjoint equation*

$$-\dot{\underline{\lambda}}^*(\tau) = \frac{\partial H(\underline{\lambda}^*(\tau), \underline{x}^*(\tau), \underline{u}^*(\tau), \tau)}{\partial \underline{x}(\tau)} \qquad (2.5)$$

Finally, it is necessary that the control, $\underline{u}$, globally minimizes the Hamiltonian for all time, $\tau \in [\tau_0, \tau_f]$, such that for problems with no constraints on the controls, the *optimality equation* is

$$0 = \frac{\partial H(\underline{\lambda}^*(\tau), \underline{x}^*(\tau), \underline{u}^*(\tau), \tau)}{\partial \underline{u}(\tau)} \qquad (2.6)$$

and for problems with constraints on the controls, a static optimization problem is formed where by the Hamiltonian is minimized with respect to the control at each instant of time (i.e., "point-wise").

$$\begin{cases} \min\limits_{u} H(\underline{\lambda}, \underline{x}, \underline{u}, \tau) \\ \quad s.t. \quad \underline{u} \in \mathbf{U} \end{cases} \tag{2.7}$$

where $\mathbf{U}$ is the set of allowable controls (i.e., the *admissible set*). This is referred to as the *Hamiltonian Minimization Condition (HMC)* and is based on *Pontryagin's Minimum Principle* [118] such that

$$H(\lambda^*(\tau), x^*(\tau), u^*(\tau), \tau) \le H(\lambda^*(\tau), x^*(\tau), u(\tau), \tau), \quad \forall \tau \in [\tau_0, \tau_f] \tag{2.8}$$

whereby u^* is the *extremal control* solution sometimes symbolically expressed as

$$\underline{u}^*(\cdot) = \arg \min_{u \in U} H(\underline{\lambda}, \underline{x}, \underline{u}, \tau) \tag{2.9}$$

Note that depending on the sign convention in the definition of the Hamiltonian, this is sometimes referred to as the maximum principle [80],[82].

Depending on specific boundary conditions (free vs. fixed), an additional set of transversality conditions are required that incorporates the endpoint constraints and the Mayer term in the cost function. The general *Transversality Condition (TC)* is given by

$$\begin{aligned} &\left[\frac{\partial \overline{E}(x^*(\tau_f), \tau_f)}{\partial \underline{x}(\tau_f)} - \lambda^*(\tau_f) \right] \delta \underline{x}_f \\ &\quad + \left[H(\lambda^*(\tau_f), x^*(\tau_f), u^*(\tau_f), \tau_f) + \frac{\partial \overline{E}(x^*(\tau_f), \tau_f)}{\partial \tau} \right] \delta \tau_f = 0 \end{aligned} \tag{2.10}$$

For example, if τ_f were specified (i.e., "fixed" final time), then $\delta \tau_f = 0$ and $\delta \underline{x}_f$ is arbitrary such that the costate is "transversal" to the terminal manifold, $e(\underline{x}_f, \tau_f)$; therefore, Eq. (2.10) reduces to the *Terminal Transversality Condition (TTC)*

$$\underline{\lambda}(\tau_f) = \frac{\partial \overline{E}}{\partial \underline{x}_f} \tag{2.11}$$

where $\overline{E}$ is the *Endpoint Lagrangian* given by

$$\overline{E}(\underline{\upsilon}, \underline{x}_f, \tau_f) := E(\underline{x}_f, \tau_f) + \underline{\upsilon}^T e(\underline{x}_f, \tau_f) \tag{2.12}$$

and similar to λ in the Hamiltonian formulation, Eq. (2.3), υ is the Lagrange multiplier associated with the endpoint constraint $e(\underline{x}_f,\tau_f)=0$.

On the other hand, if τ_f were not specified (i.e., “free” final time), then $\delta\underline{x}_f=0$ and $\delta\tau_f$ is arbitrary such that Eq. (2.10) reduces to the *Hamiltonian Value Condition (HVC)*

$$H(\tau_f)+\frac{\partial\overline{E}}{\partial\tau_f}=0 \tag{2.13}$$

where $H(\tau_f)\equiv H(\lambda^*(\tau_f),x^*(\tau_f),u^*(\tau_f),\tau_f)$.

To indicate the nature of the Hamiltonian with respect to time and in accordance with the Minimum Principle, the minimized Hamiltonian evolves according to the *Hamiltonian Evolution Equation (HEE)*

$$\dot{H}=\frac{\partial H}{\partial\tau} \tag{2.14}$$

Omitting the states, costates, and control’s dependence on time (i.e., $f(\tau)$) for simplicity of notation, the totality of necessary conditions for optimality are succinctly written as

$$\boxed{\begin{array}{c} H(\underline{\lambda},\underline{x},\underline{u},\tau)=F(\underline{x},\underline{u},\tau)+\underline{\lambda}^T f(\underline{x},\underline{u},\tau) \\ \dot{\underline{x}}=\dfrac{\partial H}{\partial\underline{\lambda}} \\ -\dot{\underline{\lambda}}=\dfrac{\partial H}{\partial\underline{x}} \\ \dot{H}=\dfrac{\partial H}{\partial\tau} \\ H(\lambda^*,x^*,u^*,\tau)\le H(\lambda^*,x^*,u,\tau) \\ \underline{\lambda}(\tau_f)=\dfrac{\partial\overline{E}}{\partial\underline{x}_f} \\ H(\tau_f)+\dfrac{\partial\overline{E}}{\partial\tau_f}=0 \\ \text{where }\ \overline{E}(\underline{\upsilon},\underline{x}_f,\tau_f):=E(\underline{x}_f,\tau_f)+\underline{\upsilon}^T e(\underline{x}_f,\tau_f) \end{array}} \tag{2.15}$$

and with the appropriate boundary conditions this system of equations can be solved for the optimal states, costates, and controls. Solving this system amounts to solving a two-point boundary value problem (TPBVP) where the initial conditions of the states are usually known and the final conditions are determined from transversality, Eq. (2.10). Now, given Eqs. (2.3)-(2.14), the Minimum Principle is summed up by Theorem 1.

<u>Theorem 1</u>: *<u>The Minimum Principle</u>* [109] Given an optimal solution $\left\{\mathbf{x}^*(\cdot), \mathbf{u}^*(\cdot), t_f^*\right\}$ to the problem, there exists a costate, $\boldsymbol{\lambda}(\cdot)$, and a covector, $\mathbf{v}(\cdot)$, that satisfies the Adjoint Equation, the HMC, the HVC, the HEE, and the Transversality Condition.

Elaborating on the case of bounded controls (i.e., controls subject to inequality constraints), the sub-problem of minimizing the Hamiltonian, see Eq. (2.7), is solved by applying the Karush-Kuhn-Tucker (KKT) Theorem. This consists of taking the Lagrangian of the Hamiltonian:

$$\overline{H}(\ldots,u,\mu) = H(\cdot) + \mu^T h \tag{2.16}$$

where μ is a KKT multiplier and h is the control constraint vector such that the appropriate necessary condition becomes:

$$\frac{\partial \overline{H}}{\partial \underline{u}} = \frac{\partial H}{\partial \underline{u}} + \left(\frac{\partial h}{\partial \underline{u}}\right)^T \mu = 0 \tag{2.17}$$

Also, the multiplier-constraint pair must satisfy the following KKT *Complementarity Conditions (CC)*:

$$\mu \begin{cases} \leq 0 & if \quad u = u^L \\ = 0 & if \quad u^L < u < u^U \\ \geq 0 & if \quad u = u^U \end{cases} \Rightarrow \quad u = \begin{cases} u^L & if \quad \mu \leq 0 \\ u^U & if \quad \mu \geq 0 \\ \end{cases} \tag{2.18}$$

4. The Generalized Karush-Kuhn-Tucker (KKT) Theorem

The KKT theorem is the primary theorem used when solving nonlinear parameter optimization problems and in particular, the Hamiltonian minimization problem, given by Eq. (2.7). In the typical NLP notation, according to Refs. [120] and [121], the general case NLP problem is stated as such:

NLP Problem (P):

$$\min \; f(x)$$
$$s.t. \begin{cases} g_i(x) \leq 0, \;\; \text{for } i = 1,...,I \quad (u_i) \\ h_j(x) = 0, \;\; \text{for } j = 1,...,J \quad (v_j) \\ \qquad\qquad x \in \mathbb{R}^n \end{cases} \tag{2.19}$$

where the dual variables are represented as u_i and v_j (i.e., Lagrange multipliers). The corresponding KKT necessary optimality conditions for problem (P) are then

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^{I} u_i^* \nabla g_i(x^*) + \sum_{j=1}^{J} v_j^* \nabla h_j(x^*) = 0 & \text{"Stationary Condition"} \\ \left. \begin{array}{l} g_i(x^*) \leq 0, \; \forall i = 1,...,I \\ h_j(x^*) = 0, \; \forall j = 1,...,J \end{array} \right\} & \text{"Primal Feasibility (PF) Condition"} \\ \left. \begin{array}{l} u_i^* g_i(x^*) = 0, \; \forall i = 1,...,I \\ \quad u_i^* \geq 0, \; \forall i = 1,...,I \end{array} \right\} & \text{"Complementary Slackness (CS) Condition"} \\ v_j \; \textit{unrestricted}, \; \forall j = 1,...,J & \text{"Dual Feasibility (DF) Condition"} \end{cases} \tag{2.20}$$

where the stationary condition is valid for n-nonlinear equations, the PF condition is valid for I-nonlinear inequalities and/or J-nonlinear equalities, and the CS condition is only valid for inequalities. These generalized KKT conditions collectively represent the same necessary conditions posed in the OCP.

It is often the case that direct methods cannot be solved analytically and as such, requires the use of iterative numerical techniques.

5. Verification of Feasibility and Optimality

To confirm feasibility and optimality of the extremal solution, both analytical and numerical methods exist.

a. Propagation Test for Feasibility

After confirming that constraints are not violated, feasibility of a computational solution can be verified by simply propagating the equations of motion with the extremal control and then comparing the DIDO state history with the propagated state history as illustrated in Figure 2.1.

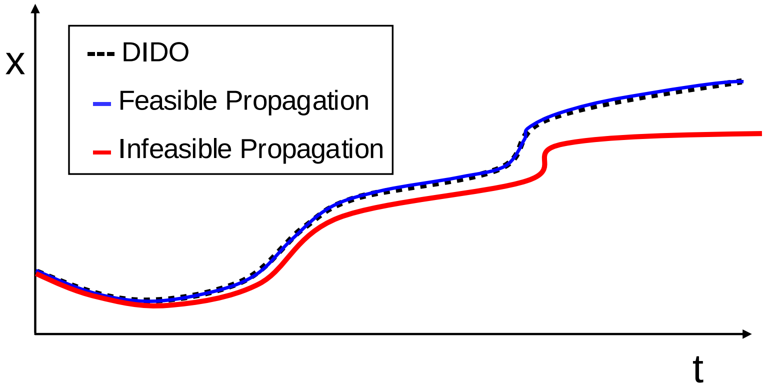


Figure 2.1 Illustration of Feasibility Test via Propagation.

If the trajectories agree to within some acceptable error tolerance (zero error cannot be achieved with infinite precision), then the solution is feasible. For simple problems, it may be possible to analytically verify that all the necessary conditions are satisfied (i.e., KKT conditions establish primal feasibility, dual feasibility, etc.) Note however, that to guarantee optimality, second-order sufficiency conditions are required. In the absence of costate information, application of Bellman's Principle is effective at numerically confirming optimality.

b. Bellman Test for Optimality

Bellman's Principle of Optimality is a simple, yet powerful tool for use in control applications. This principle essentially states that by using any point on the original optimal trajectory as an initial condition to a new problem, with all other problem formulation parameters the same, should result in the same optimal trajectory with the same or better cost. In the context of Figure 2.2, given an optimal trajectory from a point A to a point B, then the trajectory to point B from a point C lying on the optimal trajectory is also optimal [109]. Applying this principle to determine the "optimal policy" for a decision process [106] is what Bellman called dynamic programming, an extension of Hamilton-Jacobi theory.

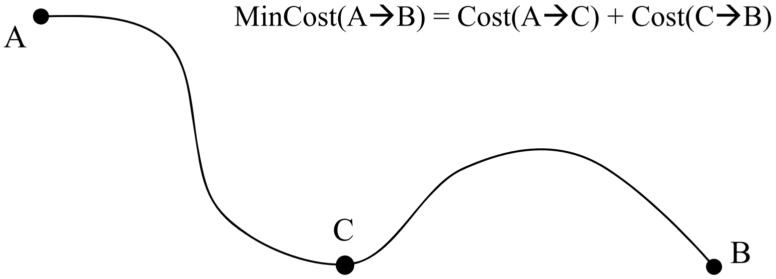


Figure 2.2 Illustration of the Principle of Optimality.

The direct application of Bellman's Principle serves many purposes throughout the work presented herein. Perhaps the most important use of the principle is to validate optimality by recalculating trajectories from intermediate points on the trajectory to ensure that they overlap the original optimal trajectory. Also, this principle is a pillar of the closed-loop, PS-feedback method discussed in Sec. II.C.2 for three primary reasons. First, it allows the use of previous optimal solutions for guesses to subsequent trajectories; hence, providing improved computational time. As discussed later, computational speed plays a key role in the successful implementation of the PS-feedback method. Second, the principle can be used to evaluate solution accuracy by increasing the number of discretization points and comparing the trajectory to a solution using fewer points. Finally, Bellman's Principle can be incorporated into an anti-aliasing technique (e.g., a^2B -algorithm) [122] that captures the high-accuracy results associated with increasing discretization points (see Sec. II.C.2.D). Evidently, this idea of discretization points is a fundamental aspect of the solution methodology discussed next.

C. SOLUTION METHODOLOGY

1. Polynomial (Legendre) PS Method

This section describes the specific polynomial-based PS method used for solving the NL OCP presented in the preceding section. Note that the following description and notation of this method was adapted from Refs. [109], [119], [123], and the references therein.

a. Overview

The basic premise of a PS method is to discretize the NL OCP, restated in Eq. (2.21) for completeness, and approximate the states, co-states and control variables by use of Lagrange interpolating polynomials given as $L_N(t)$ where the unknown coefficient values coincide with the Legendre-Gauss-Lobatto (LGL) node points. These interpolating nodes are more appropriately called quadrature points given by t_j for $j = 0,...,N$ and are distributed on the interval [-1, 1] where $t_0 = -1$ and $t_N = 1$ and for $1 \leq j \leq N-1$, t_j are the zeros of the Legendre derivative, $\dot{L}_N(t)$.

$$\min_u J\left(\underline{x}(\cdot),\underline{u}(\cdot),\tau_0,\tau_f\right) = E\left(\underline{x}(\tau_0),\underline{x}(\tau_f),\tau_0,\tau_f\right) + \int_{\tau_0}^{\tau_f} F\left(\underline{x}(\tau),\underline{u}(\tau),\tau\right) d\tau$$

$$\text{subject to} \begin{cases} \dot{\underline{x}}(\tau) = \underline{f}\left(\underline{x}(\tau),\underline{u}(\tau),\tau\right) \\ \underline{h}^L \leq \underline{h}\left(\underline{x}(\tau),\underline{u}(\tau),\tau\right) \leq \underline{h}^U \\ \underline{e}^L \leq \underline{e}\left(\underline{x}(\tau_0),\underline{x}(\tau_f),\tau_0,\tau_f\right) \leq \underline{e}^U \\ \underline{x}^L \leq \underline{x}(\tau) \leq \underline{x}^U \\ \underline{u}^L \leq \underline{u}(\tau) \leq \underline{u}^U \end{cases} \tag{2.21}$$

One of the advantages of using the Lagrange interpolating polynomial is that the node points do not have to be equally spaced. It is often the case that using equally spaced points leads to inaccurate approximations [124],[125]. In fact, optimal node placement occurs when the nodes are roots of global orthogonal polynomials such as the Legendre or Chebyshev (a.k.a. Tschebyscheff) polynomials as dictated by approximation theory [124]-[128]; hence, the use of LGL points in the PS method. For example, Figure 2.3 shows the tendency of arbitrary spaced LGL node points to cluster at the end-points.

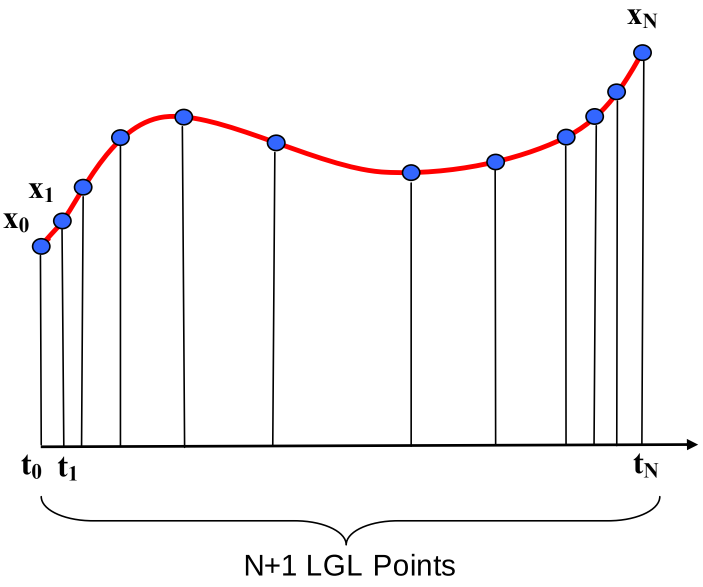


Figure 2.3 Optimal LGL Nodal Point Locations

It is also of interest to note that the PS method is equivalent to a Galerkin method with the addition of quadrature as defined in Ref. [129] as "collocation-at-the-Gaussian-quadrature-abscissa."

Now, in order to map time from the computational domain, $t \in [-1,1]$, to the physical domain, $\tau \in [\tau_0, \tau_f]$, (i.e., $\tau \mapsto t$) the following transformation

$$\tau(t) = \frac{(\tau_f - \tau_0)t + (\tau_f + \tau_0)}{2} \tag{2.22}$$

and its derivative

$$\frac{d\tau}{dt} = \frac{\tau_f - \tau_0}{2} \tag{2.23}$$

are used in the discretization of the OCP, Eq. (2.21). After the domain transformation, the continuous state and control variables, respectively, are approximated by using N^{th}-degree polynomials of the form

$$x(t) \approx x^N(t) = \sum_{j=0}^{N} x_j \phi_j \tag{2.24}$$

$$u(t) \approx u^N(t) = \sum_{j=0}^{N} u_j \phi_j \tag{2.25}$$

where, for $j = 0,1,...,N$,

$$\phi_j(t) = \frac{1}{N(N+1)L_N(t_j)} \frac{(t^2-1)\dot{L}_N(t)}{t-t_j} \tag{2.26}$$

are the Lagrange interpolating polynomials of order N such that

$$\phi_j(t_k) = \delta_{jk} = \begin{cases} 1 \text{ if } j = k \\ 0 \text{ if } j \neq k \end{cases} \tag{2.27}$$

where δ_{jk} is the Kroneker delta.

When using Eqs. (2.24)-(2.26), the approximation is actually taking place between the nodes since the values at the t_k points are exactly equal to the approximated values.

The derivative of this approximating polynomial at the collocation points t_k results in a matrix multiplication in the following form

$$\underline{\dot{x}}^N(t_k) = \sum_{j=0}^{N} \underline{x}_j \dot{\phi}_j(t_k) = \sum_{j=0}^{N} D_{kj}\underline{x}_j \tag{2.28}$$

where $D_{kj} = \dot{\phi}_j(t_k)$ are the LGL points and are the entries of the $(N+1)\times(N+1)$ differentiation matrix **D** given as,

$$\mathbf{D} := [D_{kj}] := \begin{cases} \dfrac{L_N(t_k)}{L_N(t_j)} \cdot \dfrac{1}{t_k - t_j} & k \neq j \\[2ex] -\dfrac{N(N+1)}{4} & k = j = 0 \\[2ex] \dfrac{N(N+1)}{4} & k = j = N \\[2ex] 0 & \text{otherwise} \end{cases} \tag{2.29}$$

In this form, and using the linear mapping between the computational and physical domains, the derivatives of the state approximations satisfy the differential equations exactly at the node points as given by the condition

$$\underline{\dot{x}}^N(t_k) = \frac{\tau_f - \tau_0}{2}\underline{f}\left(\underline{x}_k, \underline{u}_k, \tau_k\right) \tag{2.30}$$

Remarks: The PS D-matrix (2.29) works similar to that of a finite difference matrix in approximating derivatives except that equispaced data points are not required. Also, a worthwhile discussion on the accuracy of Legendre spectral derivatives can be found in Ref. [119].

The strength of the D-matrix comes into play by the computational benefits of its sparsity structure. Sparsity in the Jacobian matrix reduces the sensitivity in the BVP [115]. A review of sparsity patterns and associated computational benefits can be found in Ref. [130]. Also, depending on the problem, it is possible that a low number of discretization points can still provide relatively accurate solutions [131]-[133].

In addition to the derivative approximation, integral approximation is necessary for the integral portion of the cost function stated in Eq. (2.21). Since Gauss quadrature results in superior performance over other integral approximation rules [125],[134] and to include the interval endpoints (-1 and 1) in the computational domain, a Gauss-Lobatto integration ruleis applied. The general Gaussian quadrature rule expressed as

$$\int_{-1}^{1} f(t)dt \approx \sum_{k=1}^{n} w_k f(t_k) \tag{2.31}$$

for n points is transformed to its corresponding domain with the mapping $t \to \tau$ such that

$$\int_{\tau_0}^{\tau_f} f(\tau)d\tau = \frac{\tau_f - \tau_0}{2}\int_{-1}^{1} f(\tau(t))dt \tag{2.32}$$

Therefore, the cost function in Eq. (2.21) is transformed to a discretized form,

$$J[\cdot] \approx J^N[\hat{x}, \hat{u}, \tau_0, \tau_f] = E(x_0, x_N, \tau_0, \tau_f) + \frac{\tau_f - \tau_0}{2}\sum_{k=0}^{N} F(x_k, u_k, \tau_k)w_k \tag{2.33}$$

where $\hat{x} = (x_0, x_1, \ldots x_N)$, $\hat{u} = (u_0, u_1, \ldots u_N)$ and w_k are the LGL weights given by

$$w_k := \frac{2}{N(N+1)}\frac{1}{[L_N(t_k)]^2}; \quad k = 0, 1, \ldots, N. \tag{2.34}$$

In addition to the discretization of the cost function, the mixed state-control path constraints and the end-point constraints can likewise be evaluated at the LGL node points to form a complete discretized representation of the OCP.

b. Covector Mapping Principle and Costate Estimation

Bridging from a theorem developed in Refs. [128] and [135], the Covector Mapping Principle (CMP) has become instrumental in the automation of verifying the necessary conditions for optimality. It entails the application of the KKT theorem to the NLP problem such that the KKT multipliers of the solved NLP elegantly map to the discrete costates. In other words, it can be shown that the Lagrange multipliers (duals) for the NLP problem map to the costate values at the discrete LGL points [128]. The Lagrange multipliers are numerically determined by collocating the costate differential equations at t_k along with the transversality conditions and the optimality condition to obtain a system of linear equations that can be solved for the Lagrange multipliers. A thorough overview of the historical introduction of this principle is found in Ref. [136] with recent extensions to solving nonlinear optimal control problems [137]. In addition to the CMP providing a means to verify optimality conditions, it is also instrumental in establishing the convergence properties of PS methods.

c. Convergence

Convergence of discretization methods for optimal control is still a very active area of research and until recently, a more rigorous proof of PS-convergence has been unavailable. A series of recent developments in the theoretical foundations for PS-convergence are addressing people's concern [138]-[140]. For example, Gong et al., exploits the normal form of feedback linearizable dynamics to modify the standard PS method (similar to dynamic inversion); thereby facilitating the proof of sufficient conditions for the existence and convergence of PS discretizations [141],[142].

d. NLP Solver used for DIDO's PS Method

SNOPT is a collection of Fortran 77 subroutines that uses the iterative SQP method to solve the NLP problem [103]. SQP is one the most widely used algorithms for solving general NLP problems. It is a recursive program that uses Newton's Method to directly solve the KKT conditions such that the sub-problem minimizes a quadratic approximation to the Lagrangian function that is optimized over a linear approximation of the constraints. As such, this method is also referred to as the Lagrange-Newton, or projected Lagrangian approach [120].

e. Benefits of PS Method

The primary benefits that this method provides over other methods are the following:

1. discretization ideally reduces problem from large-scale to small-scale via sparsity structure; hence, providing high computational speed;
2. when a solution exists, global convergence is guaranteed;
3. CMP automatically provides dualization of problem variables;
4. can solve non-smooth problems [143]; and
5. "black-box" implementable using simplistic DIDO software package.

Because of this method's ability to provide real-time (or near real-time depending on the problem), open-loop optimal solutions, feedback control is possible as explained in the next section.

2. PS-Feedback Method

a. Overview

The confluence of recent developments in optimal control theory, numerical methods and computational power has provided a turning point in the development of non-traditional, nonlinear feedback control laws. Stemming from the underlying principles of sample and hold feedback control, a PS-feedback algorithm has been effectively implemented in recent years to solve various guidance, navigation, and control problems that include a non-Eulerian time-optimal spacecraft slew maneuver [144], nonlinear observer [145], stabilization through real-time control [146], optimal reentry guidance [147], and control of a classical inverted pendulum [146] to name just a few applications. The proceeding section summarizes the recent theoretical foundations described in Ref. [148] to support the notion of real-time as facilitated by a key lemma that links the Lipschitz constant of the dynamics to the requirements of a sampling frequency for generating Carathéodory-π solutions.

b. Carathéodory-π Solutions

The solution to a general nonlinear system, $\dot{x} = f(x,u,t)$, with state and control spaces represented by $x(t) \in \mathbf{X}(t)$ and $u(t) \in \mathbf{U}(t, x(t))$, respectively, that satisfies certain "C^1-Carathéodory" conditions supports the concept of using π-solutions to generate feedback maps [148]. The notion of π-solutions has to do with partitioning the time interval according to a piecewise series of open-loop control solutions such that when executed in sequence are equivalent to a closed-loop solution. This is possible since any digital implementation of a closed-loop continuous-time system is fundamentally a series of discrete signals governed by the sampling frequency of that system. Note that an important consideration for practical flight control is that the control space is state dependent [47]. In addition to generating feedback maps, this approach also requires the control trajectory satisfy an optimality criterion and that the initial and final states meet some specified endpoint conditions defined by the problem.

To implement this sample-and-hold strategy, consider the following example, restated from Ref. [148].

Example: Given an initial condition, x_0, a π-trajectory, $x(\cdot)$, corresponding to a time-invariant feedback policy, $u = k(x)$, is generated by the following:

From t_0 to t_1, generate a solution to

$$\dot{x}(t) = f(x(t), k(x_0)), \quad x(t_0) = x_0, \quad t \in [t_0, t_1] \tag{2.35}$$

Then, at $t = t_1$, set $x_1 = x(t_1)$ and restart the system with $u = k(x_1)$ such that

$$\dot{x}(t) = f(x(t), k(x_1)), \quad x(t_1) = x_1, \quad t \in [t_1, t_2] \tag{2.36}$$

and continue this process to form a set of piecewise open-loop solutions (i.e., a π-solution).

This same procedure would work for a time-varying feedback law, $u = k(x,t)$, but may be non-conducive for practical feedback control since it does not exploit the inherent element of prediction provided by the clock-time, *t*. Therefore, instead of basing the sampling on past data (i.e., 'backward-looking' approach) it is better to base it on future data (i.e., 'forward-looking' approach) [148] such that the control is continuously updated by the clock information. In this case, by imposing the Carathéodory conditions to the generation of all the segmented solutions, an optimal Carathéodory-π trajectory results.

The following section provides an overview of the practical implementation to generating these Carathéodory-π trajectories as well as the feedback-based computational time requirements.

c. *Practical Implementation and Computational Time Requirements*

Abandoning the notion of seeking analytical or closed-form expressions for feedback, $x \mapsto k(x)$, a PS-feedback method relies on an optimal feedback strategy that resorts to a computational algorithm manifesting a more fundamental control form. This method automatically generates a time-varying feedback law, $u = k(x,t)$, in a semi-discrete form. As depicted in Figure 2.4, clock-time is used to generate open-loop "analog" signals, $u(t) = k(t, x(t_i))$ over a discrete state space as facilitated by discrete sensor measurements.

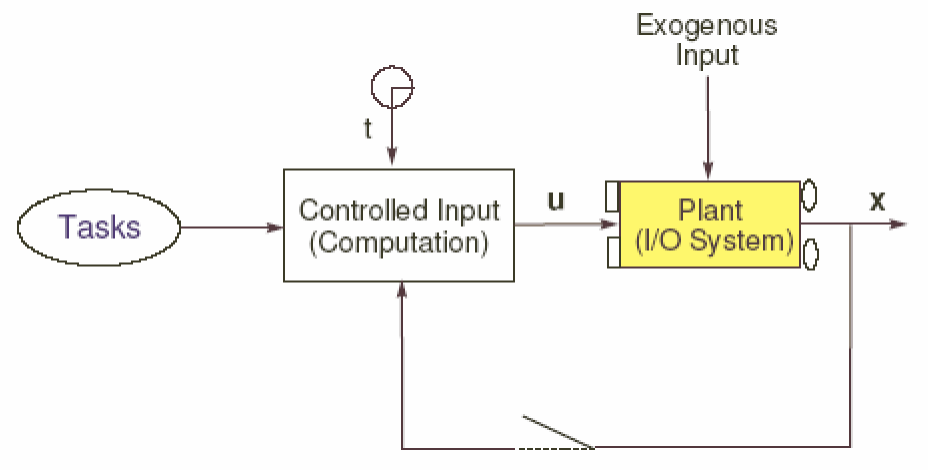


Figure 2.4 Schematic for Generating a Semi-Discrete, Clock-Based Feedback Controller [148].

Consider a nonlinear *model* of a control system

$$\dot{x} = f(x,u,t,\, p_0) \tag{2.37}$$

where $p_0 \in \mathbb{R}^{N_p}$ is a constant representing the system parameters (e.g., mass, reference area, etc). Now, let the real system (plant) dynamics be

$$\dot{x} = f(x,u,t,\, p) + d(t) \tag{2.38}$$

where p is the actual plant parameters and $d(t)$ be some exogenous input function such that a perfect model results when $p = p_0$ and $\|d\|_{L^\infty} = 0$. According to Figure 2.5, a feedback policy is implemented starting at $t = t_i$ with the computation of $[t_i, t_{i+1}] \mapsto u = k(t, x_R(t_i))$, where x_R is the state of the real system (plant).

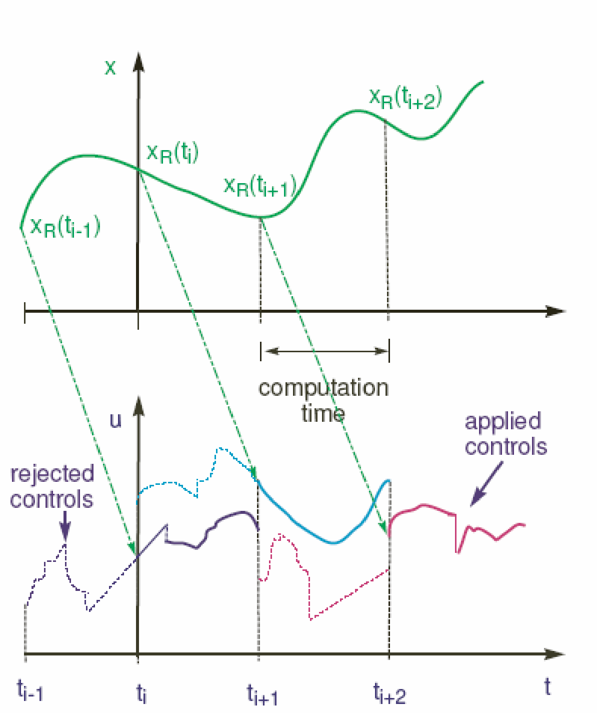


Figure 2.5 Practical Implementation for a Clock-Based Feedback Control.

Under the action of an open-loop control, $[t_i, t_{i+1}] \mapsto k(t, x_R(t_i))$, the state of the model at t_{i+1} is given by,

$$x_M(t_{i+1}) = x_R(t_i) + \int_{t_i}^{t_{i+1}} f(x_M(t), k(t, x_R(t_i)), t; p_0)dt \tag{2.39}$$

As seen in Figure 2.5, the control $t \mapsto k(t, x_R(t_i))$ is available at t_{i+1} for application to the plant. Here, the computational time is

$$\tau_i := t_{i+1} - t_i \tag{2.40}$$

Thus, the state of the plant at t_{i+1} is determined by the action of the control, $[t_i, t_{i+1}] \mapsto k(t, x_R(t_{i-1}))$, and is given by,

$$x_R(t_{i+1}) = x_R(t_i) + \int_{t_i}^{t_{i+1}} f(x_R(t), k(t, x_R(t_{i-1})), t; p)dt + \int_{t_i}^{t_{i+1}} d(t)dt \tag{2.41}$$

Therefore, the real trajectory x_R differs from the ideal/model trajectory, x_M due to the effects of the computational delay time as well as the deviations caused by disturbances, $d(t)$ and uncertainties, *p*. Subtracting (2.39) from (2.41) gives,

$$\begin{aligned} x_R(t_{i+1}) - x_M(t_{i+1}) = & \int_{t_i}^{t_{i+1}} f(x_R(t), k(t, x_R(t_{i-1})), t; p)dt \\ & - \int_{t_i}^{t_{i+1}} f(x_M(t), k(t, x_R(t_i)), t; p_0)dt + \int_{t_i}^{t_{i+1}} d(t)dt \end{aligned} \tag{2.42}$$

Now, assuming *Lipschitz* continuity*,* Eq. (2.42) can be re-written as

$$\begin{aligned} x_R(t_{i+1}) - x_M(t_{i+1}) \le & \, Lipf_x \int_{t_i}^{t_{i+1}} \left\| x_R(t) - x_M(t) \right\| dt \\ & + Lipf_u \int_{t_i}^{t_{i+1}} \left\| k(t, x_R(t_{i-1})) - k(t, x_R(t_i)) \right\| dt \\ & + Lipf_p \left\| p - p_0 \right\| \tau_i + \left\| d \right\|_{L^\infty} \tau_i \end{aligned} \tag{2.43}$$

Extending this result, Ross proves that the Lipschitz constant of a system is related to the required computational time by

$$\tau_i \le \frac{W(r)}{Lipf_x} \tag{2.44}$$

where $W(r)$ is a multi-valued Lambert W function given by $x = W(x)e^{W(x)}$ [148]. Additional remarks regarding a Lemma and its proof for this relation can be found in [148].

The key for successful implementation of these feedback principles rely on a sufficiently fast generation of open-loop controls. By generating open-loop controls as demanded by Eq. (2.44), closed loop is achieved by generating Carathéodory-π solutions [148]. In recent years, it has become quite apparent that PS methods [126],[149],[150] are capable of generating optimal open-loop controls within fractions of a second, [142],[151],[152] even when implemented in legacy hardware running MATLAB©. This implies that real-time optimal controls can be generated for systems with large Lipschitz constants; that is, systems with fast dynamics. This simply follows by re-writing (2.44) as,

$$Lipf_x \leq \frac{W(r)}{\tau_c} \tag{2.45}$$

where τ_c is the largest computational time. Chapter IV provides results that demonstrate this concept for the reentry guidance of an RLV.

As mentioned earlier, the software package DIDO, based on PS-methods, provides a means of computing optimal controls. Requiring only the problem formulation as described by Eq. (2.21), DIDO is a "minimalist's approach" to solving optimal control problems. Also, by incorporating the CMP, it allows "dualization to commute with discretization"; hence, automatically verifying necessary optimality conditions [153].

d. Algorithms for Trajectory Generation

The PS-feedback implementation essentially consists of both off-line and on-line trajectory generation via DIDO as depicted by the control architecture shown in Figure 2.6. The purpose of this architecture is to illustrate the overall system's use of an off-line optimal trajectory solution (actually an on-line "start-up" run), successive on-line optimal trajectory solutions, and a Runge-Kutta propagation scheme to integrate the equations of motion in order to determine the current vehicle state vector (idealize the real vehicle's navigation).

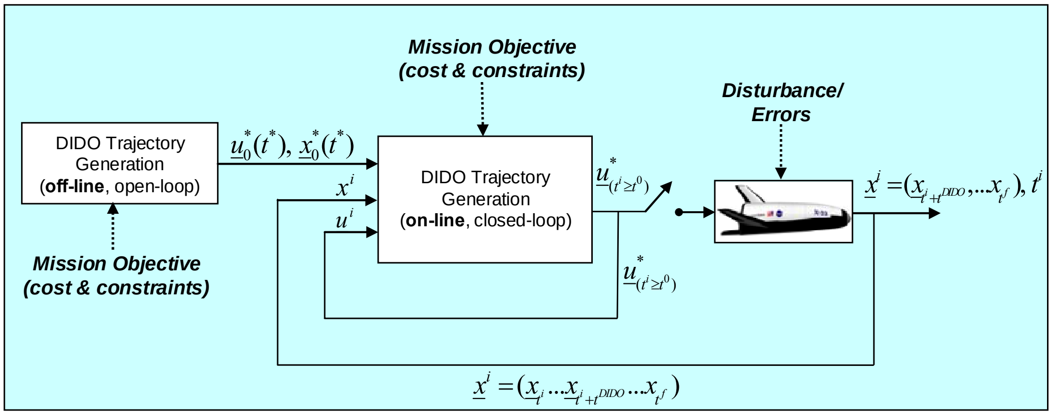

Mission Objective
(cost & constraints)
Disturbance/
Errors
DIDO Trajectory
Generation
(off-line, open-loop)
$\underline{u}_0^*(t^*), \underline{x}_0^*(t^*)$
x^i
u^i
DIDO Trajectory
Generation
(on-line, closed-loop)
$\underline{u}^*_{(t^i \geq t^0)}$
$\underline{x}^i = (\underline{x}_{t^i+t^{DIDO}},...x_{t^f}), t^i$
$\underline{u}^*_{(t^i \geq t^0)}$
Mission Objective
(cost & constraints)
$\underline{x}^i = (\underline{x}_{t^i}...\underline{x}_{t^i+t^{DIDO}}...x_{t^f})$

Figure 2.6 Control Architecture for Off-Line and On-Line Trajectory Generation.

Given initial conditions and desired final conditions, cost, and constraints (i.e., mission objectives), DIDO computes the first, open-loop optimal control. This optimal solution $(u^*(t^*), x^*(t^*))$ is used as an initial guess to begin the closed-loop implementation. The closed-loop optimal control is then solved on-line by using the open-loop solution as the start-up values to a new open-loop OCP. As the next DIDO run is in progress, the previous control history is interpolated and used to propagate the equations of motion. Upon DIDO completion, the vehicle's current state is provided from the propagation routine and a new optimal trajectory is computed using the same constraints as the start-up problem. Note that the numerical errors (e.g.,, propagation) in the computational algorithm, represented by "errors" in Figure 2.6, are associated with an imperfect model (i.e., plant uncertainties) and imperfect measurements (i.e., navigation/estimation errors). This successive process of solving the full OCP is illustrated in Figure 2.7. CAUTION: The propagated segments are exaggerated for illustration purposes only and the complete open-loop trajectory beginning from each successive initial condition (IC) is not shown.

Since the precision of real control systems is limited by the accuracy of the sensors/estimation, it is more practical to use epsilon balls, $x_i \in \mathbf{B}(x^i, \varepsilon)$, in place of initial and final point-conditions, $x_0 = x^0$ and $x_f = x^f$, respectively. The epsilon balls, indicated in Figure 2.7, are then dictated by the accuracy of the sensors/estimation or the precision requirement dictates the sensor selection. Ideally, each initial condition will begin exactly at the final condition of the previous optimal trajectory segment. This will capture the effects of Bellman's Principle as described earlier.

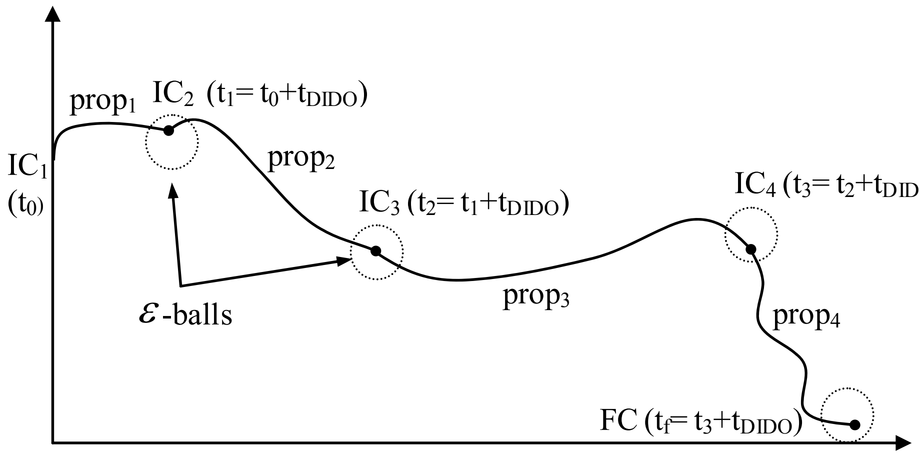

prop$_1$
IC$_2$ (t$_1$= t$_0$+t$_{DIDO}$)
IC$_1$
(t$_0$)
prop$_2$
IC$_3$ (t$_2$= t$_1$+t$_{DIDO}$)
IC$_4$ (t$_3$= t$_2$+t$_{DIDO}$)
ε-balls
prop$_3$
prop$_4$
FC (t$_f$= t$_3$+t$_{DIDO}$)

Figure 2.7 Successive Optimization Concept.

As alluded to earlier, one powerful application of Bellman's Principle is to de-alias a low-accuracy solution to provide a high-accuracy solution without creating a large-scale optimization problem. Ross et al. makes the connection between the Nyquist-Shannon sampling theorem and generic mesh refinement methods that require increasing the number of sampling points [122]. It is shown that a 15-node solution is an alias of the 60-node solution. In later chapters, this principle and the a^2B -algorithm are applied to a 6-DOF reentry problem to both test and verify the accuracy of the solution. For this reason, the a^2B -algorithm is restated here for completeness.

a^2B -algorithm: [122]

1. Solve the problem for a low number of nodes, *n*. This generates a discrete-time solution, $\{x_i, u_i\}_{i=0}^{n}$ corresponding to discrete times $\{t_i\}_{i=0}^{n}$.
2. Partition the time interval $[t_0, t_n]$ into N_B-Bellman segments, $t_0 < t^1 < \ldots < t^{N_B} = t_n$. These segments need not be uniformly spaced.
3. Propagate the differential equation from t_0 to t^1 using x_0 as the initial condition and any method of continuous-time reconstruction of the controls, $u^1(t),\ t \in [t_0, t^1]$ based on $\{u_i\}_{i=0}^{n}$. That is, solve the initial value problem,

$$\dot{x} = f(x, u^1(t)), \quad x(t_0) = x_0 \tag{2.46}$$

This step generates a continuous-time trajectory, $x^1(t),\ t \in [t_0, t^1]$. This propagation is done numerically via some high-precision propagator, say the standard 4/5 Runge-Kutta method.

4. Set $x_0 = x^1(t^1)$ and $t_0 = t^1$ and go to step 1; that is, set a new initial condition as the value of the integrated state at the end of the period, $[t_0, t^1]$ and solve the problem again for *n* (which continues to be low). This generates a new sequence $\{x_i, u_i\}_{i=0}^{n}$ corresponding to new discrete times $\{t_i\}_{i=0}^{n}$, etc.

5. The algorithm stops at the N_B-th sequence when the final conditions are met. The candidate optimal trajectory is given by the Bellman chain, $\left\{x^1(t),\ t \in [t_0, t^1];\ x^2(t),\ t \in [t^1, t^2];\ \ldots\ x^{N_B}(t),\ t \in [t^{N_B-1}, t^{N_B}]\right\} := x_B(t),\ t \in [t_0, t_f]$ Similarly, the corresponding controls are given by, $\left\{u^1(t),\ t \in [t_0, t^1];\ u^2(t),\ t \in [t^1, t^2];\ \ldots\ u^{N_B}(t),\ t \in [t^{N_B-1}, t^{N_B}]\right\} := u_B(t),\ t \in [t_0, t_f]$

To further illustrate the actual implementation logic in DIDO, the flowchart for the feedback algorithm is shown in Figure 2.8, adapted from Ref. [144].

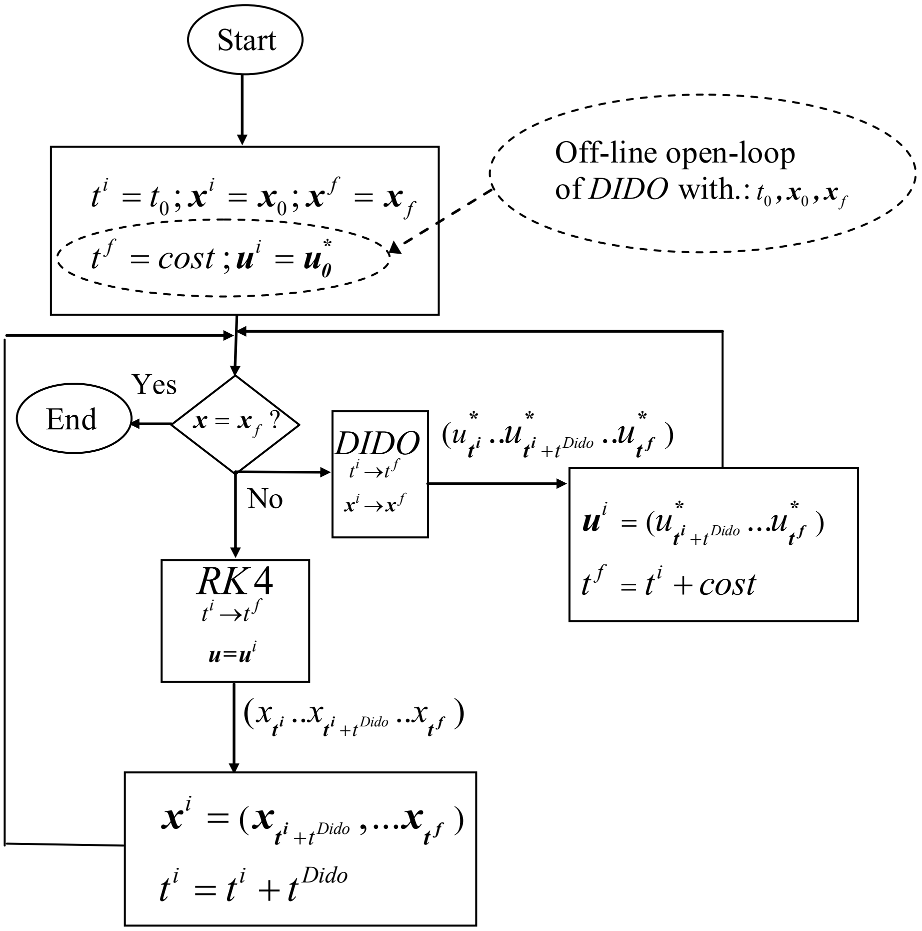

Start
$t^i = t_0; \boldsymbol{x}^i = \boldsymbol{x}_0; \boldsymbol{x}^f = \boldsymbol{x}_f$
$t^f = cost; \boldsymbol{u}^i = \boldsymbol{u}_0^*$
Off-line open-loop of *DIDO* with.: $t_0, \boldsymbol{x}_0, \boldsymbol{x}_f$
Yes
End
$\boldsymbol{x} = \boldsymbol{x}_f$?
No
DIDO
$t^i \to t^f$
$\boldsymbol{x}^i \to \boldsymbol{x}^f$
$(u^*_{t^i} .. u^*_{t^i + t^{Dido}} .. u^*_{t^f})$
$\boldsymbol{u}^i = (u^*_{t^i + t^{Dido}} ... u^*_{t^f})$
$t^f = t^i + cost$
$RK4$
$t^i \to t^f$
$\boldsymbol{u} = \boldsymbol{u}^i$
$(x_{t^i} .. x_{t^i + t^{Dido}} .. x_{t^f})$
$\boldsymbol{x}^i = (\boldsymbol{x}_{t^i + t^{Dido}}, ... \boldsymbol{x}_{t^f})$
$t^i = t^i + t^{Dido}$

Figure 2.8 Flowchart of Closed-Loop Control Algorithm [144].

e. Benefits of PS-Feedback Method

With all said, the primary benefits that this method provides over traditional feedback methods are the following:

1. no a priori knowledge of computational time is required,
2. no gains are used; hence no gain-tuning required,
3. no reference trajectory tracking since new optimal solutions are generated each sample time, and
4. no disturbance rejection/cancellation in the traditional sense (discussed later); hence, the system may autonomously use disturbance(s) as a benefit towards accomplishing the final objective.

Of course, with any numerical method, there exists computational issues that generally lead to problems in the form of numerical errors.

D. COMPUTATIONAL ISSUES

1. Numerical Problem Solving

Using numerical algorithms to solve complex, real-world problems is now the standard with the availability of significantly more powerful computers. Although this capability is appealing, it poses potential pitfalls. Primarily in the fact that numerical errors, if not identified and accounted for, can cause multiple problems to include numerical non-convergence and wrong results. For example, Hull has indicated cases where numerical errors causes non-convergence when using NLP to solve a reentry trajectory optimization problem [59]. For this reason, it is important to understand potential error sources when performing numerical computations.

2. Numerical Errors

The problem is essentially a function that maps the data to a solution [124]. The errors associated with this mapping process are primarily a result of inexact data and algorithm errors. As previously stated, using wrong data can cause errors in the true solution. Also, a numerical method using an infinite precision algorithm introduces truncation, round-off, and floating-point errors.

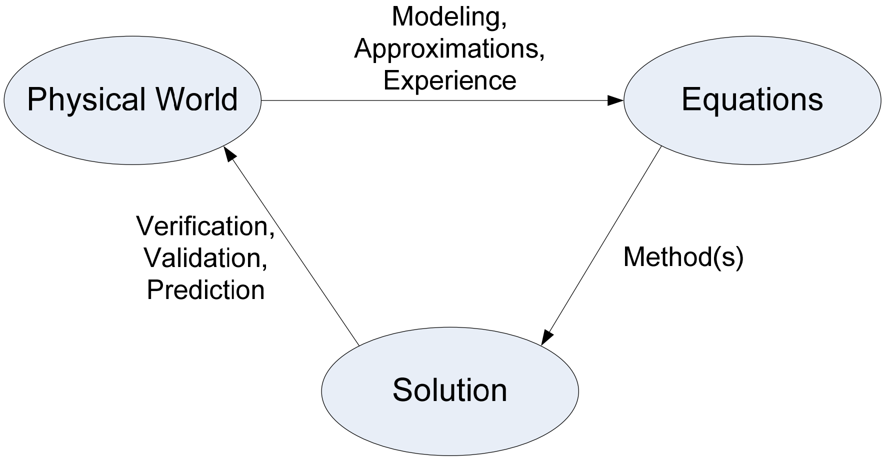


Figure 2.9 Numerical Problem Solving [124].

Most real problems are not solvable exactly. In most cases, the solution requires further approximations and/or iterations such that the exact solution only exists in the limit as the iteration number approaches infinity. In this situation, the use of computers makes the most sense. To solve a real-world problem, usually the problem is represented by equations and data. As indicated in Figure 2.9, equations require some form of approximating the physical world. Also, the data, when available, is almost always wrong due to measurement errors, floating point conversions, previous complications, etc. Ross eloquently sums this up by posing the following lemma which he calls the "Fundamental Law of Modeling" [154].

Lemma: [154] Almost all models are inexact; the only important issue is the degree of approximation.

3. Exact vs. Approximate Problems and Solutions

Unfortunately, there is rarely a clear distinction between approximately solving an exact problem and exactly solving an approximate problem. The key is in modeling or approximating the real problem (i.e., physical world) as close to exact as possible with the underlying stipulation that no matter how exact the model is, it will never be perfect. Since numerous errors already exist in numerical solvers, these errors are compounded when using a poorly modeled problem formulation.

Since high precision usually comes at a high cost, problem simplifications are almost always performed, especially in control theory. So, the question becomes: what degree of fidelity is appropriate?

4. Model Fidelity

Increasing the model fidelity of a problem usually implies that the order of the system will increase; hence, creating a large-scale OCP. The concern about the size of a problem has spawned the phrase – "curse of dimensionality" [81]. Some claim that solutions to large-scale OCPs are not possible, or not fast enough for certain applications. For example, Mettler states that high-fidelity models are "computationally too complex to be solved in real time" [155]. This may have been the case as little as 10 years ago, but since then, computational power and numerical methods have made significant improvements.

Is size and/or complexity really a problem for modern optimization algorithms? Over the past few years, PS methods have proven to be capable of solving complex, large-scale problems including trajectory optimization of reentry vehicles [46],[54], [83],[84], launch vehicle ascent guidance [123], footprint generation for RLVs [86], multiple satellite formations [156], and real-time spacecraft slew maneuvers [157] to name a few. Theoretically, it is also possible that size, can actually improve optimization performance. For example, adding more constraints may reduce the feasible region for the problem solution; hence, reducing the search space for the optimizer.

5. Problem Formulation

The key to good problem formulation is constructing a well-posed problem. General requirements of a well-posed problem are that the solution (1) exists, (2) is unique, and (3) depends continuously on the data. The requirement for uniqueness depends on the specific problem because even some well-posed problems may have non-unique solutions, but the key is to make a problem numerically tractable which is sometimes difficult with non-unique solutions. The third requirement has to do with the formulation being continuously differentiable such that the partial derivatives of the function with respect to each independent variable exist. This implies that the value of the data is important as well. For example, small changes in the data should produce only small changes in the solution for a well-posed problem. If small changes in the data results in no solution or multiple solutions, then an alternate problem formulation should be considered.

6. Scaling and Balancing

As with all numerical algorithms, scaling is critical for optimum computational performance (i.e., accuracy, speed, etc). It is especially so in optimal control problems as the trajectory variables can have substantially different magnitudes that when used in numerical computations, can adversely affect the solution. Often the case, "poor scaling can make a good algorithm bad" [115]. Ross discusses some of the effects of scaling in Ref. [158]. Good numerical practice frequently involves scaling the state, control, and time variables to a range between 0 and 1, or at least within the same order of magnitude. This helps prevent numerical imprecision that can lead to ill-conditioned matrices.

Since the intent of scaling and balancing is to provide numerically tractable quantities in the algorithms to improve computational performance, the mathematical premise is to normalize the problem variables by use of scaling factors. For example, to transform a dynamic system $\dot{x} = f(x,u,t)$ to a scaled system $\dot{\hat{x}} = f(\hat{x},\hat{u},\hat{t})$, that is,

$$\dot{x} = \frac{dx}{dt} = f(x,u,t) \;\Rightarrow\; \dot{\hat{x}} = \frac{d\hat{x}}{d\hat{t}} = f(\hat{x},\hat{u},\hat{t}) \tag{2.47}$$

First, scale the state variable(s), the control variable(s), and the time according to

$$\hat{x} = k_x x, \quad \hat{u} = k_u u, \quad \hat{t} = k_t t \tag{2.48}$$

where the scale factors k_i can be represented as $k_x = 1/XU$, $k_u = 1/UU$, $k_t = 1/TU$ with *XU, UU,* and *TU* some user defined units for the given quantities.

Next, substitute the scaled quantities of Eq. (2.48) into Eq. (2.47), to produce a scaled and balanced dynamical equation,

$$\frac{d\hat{x}}{d\hat{t}} = \frac{d(k_x x)}{d(k_t t)} = \frac{k_x}{k_t}\frac{dx}{dt} = \frac{k_x}{k_t} f(x,u,t) = \frac{k_x}{k_t} f\left(\frac{\hat{x}}{k_x}, \frac{\hat{u}}{k_u}, \frac{\hat{t}}{k_t}\right) \tag{2.49}$$

Therefore, for multiple states and controls, Eq. (2.47) is expanded to

$$\begin{gathered}
\frac{d\hat{x}_1}{d\hat{t}} = \frac{k_{x_1}}{k_t} f\left(\frac{\hat{x}_1}{k_{x_1}}, \frac{\hat{x}_2}{k_{x_2}}, \ldots, \frac{\hat{x}_i}{k_{x_i}}, \frac{\hat{u}_1}{k_{u_1}}, \frac{\hat{u}_2}{k_{u_2}}, \ldots, \frac{\hat{u}_j}{k_{u_j}}, \frac{\hat{t}}{k_t}\right) \\
\frac{d\hat{x}_2}{d\hat{t}} = \frac{k_{x_2}}{k_t} f\left(\frac{\hat{x}_1}{k_{x_1}}, \frac{\hat{x}_2}{k_{x_2}}, \ldots, \frac{\hat{x}_i}{k_{x_i}}, \frac{\hat{u}_1}{k_{u_1}}, \frac{\hat{u}_2}{k_{u_2}}, \ldots, \frac{\hat{u}_j}{k_{u_j}}, \frac{\hat{t}}{k_t}\right) \\
\cdot \\
\cdot \\
\frac{d\hat{x}_i}{d\hat{t}} = \frac{k_{x_i}}{k_t} f\left(\frac{\hat{x}_1}{k_{x_1}}, \frac{\hat{x}_2}{k_{x_2}}, \ldots, \frac{\hat{x}_i}{k_{x_i}}, \frac{\hat{u}_1}{k_{u_1}}, \frac{\hat{u}_2}{k_{u_2}}, \ldots, \frac{\hat{u}_j}{k_{u_j}}, \frac{\hat{t}}{k_t}\right)
\end{gathered} \tag{2.50}$$

where $i = 1,\ldots,N_x$ and $j = 1,\ldots,N_u$. With this result, the scale factors k_x, k_u, and k_t can be tuned to simultaneously satisfy the following general guidelines for a well-scaled problem:

1. Scale the independent variables to within the same range, e.g.,

$$(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) \in [0,1]$$

2. Scale the dependent functions to have the same order of magnitude,

$$\frac{d\hat{x}_1}{d\hat{t}} \sim O(\varepsilon),\ \frac{d\hat{x}_2}{d\hat{t}} \sim O(\varepsilon),\ \ldots\ ,\ \frac{d\hat{x}_n}{d\hat{t}} \sim O(\varepsilon)$$

It has also been discovered that scaling the dependent functions such that the costates (i.e., Lagrange multipliers) are close to the same magnitude is important as well; therefore,

3. Scale the Lagrange multipliers to have the same order of magnitude,

$$\lambda_1 \sim O(\varepsilon),\ \lambda_2 \sim O(\varepsilon),\ \ldots\ ,\lambda_n \sim O(\varepsilon)$$

A common way to normalize or scale problems is to design units based on the fundamental units of distance, mass, and time (i.e., canonical units) defined here as *DU*, *MU*, and *TU*, respectively. Like Eq. (2.48), a scaled variable is formed by applying scale factors based on these fundamental units. For example, time (*t*) and distance (*x*) can be scaled according to

$$\hat{t} \mapsto \frac{t}{TU}; \quad \hat{x} \mapsto \frac{x}{DU} \tag{2.51}$$

Likewise, velocity is distance divided by time so it can be scaled according to

$$\hat{V} \mapsto \frac{V}{VelU} = \frac{V}{DU/TU} \tag{2.52}$$

Therefore, any variable can be written in terms of the fundamental units and scaled accordingly. Note, however, that for problems involving more than one variable defined by distance units, where the relative range of each distance unit is significantly different, then each variable may require a different scale factor, *DU*. For example, consider a reentry problem using Cartesian coordinates where the crossrange is only 500 ft and the downrange is 4,000,000 ft. Obviously, it is more appropriate to scale each of these variables by different distance units. This goes back to the general form given by Eq. (2.49), implying that each variable can have a unique scale factor. For a reentry problem using spherical coordinates, this is not an issue since crossrange and downrange are in latitude and longitude units. In this case, it is common practice to scale the geographical position by degrees; altitude by the radius of the earth (or initial altitude) R_0; velocity by $\sqrt{g_0 R_0}$ where g_0 is the gravitational acceleration; and time by $\sqrt{R_0 / g_0}$ [31]. However, instead of using R_0 or arbitrarily selecting the *DU* as done in Ref. [46], it may be better in terms of normalization to base the selection of *DU* on the range of a given distance variable and a "shifting term" as recommended by Betts in Ref. [115]. That is, given a range on a variable, $x^L \le x \le x^U$, Eq. (2.48) becomes

$$\hat{x} = k_x x + s \tag{2.53}$$

where

$$k_x = \frac{1}{x_U - x_L}; \quad s = \frac{1}{2} - \frac{x_U}{x_U - x_L} \tag{2.54}$$

Although this procedure may be physically appealing, it may not be suitable for certain problems. Therefore, as illustrated by an example in Ref. [159], care must be given as to how the scaling method is affecting the computations.

Overall, scaling is problem dependent and as such, there are no set rules. Not only are the expected values of certain variables important, but so is the problem formulation itself. As implied by the previous example of coordinate system selection and demonstrated in the next chapter, careful selection of the state variables can simplify the scaling process.

E. CONCLUDING REMARKS

This section has provided a mathematical framework for the theoretical foundations of optimal control theory and the solution methodology applied in this research. The following chapters will show how the theory and the method go hand-in-hand in solving perhaps one of the most challenging aerospace control problems since the inception of the space era – the guidance and control for the atmospheric reentry of a reusable launch vehicle.

THIS PAGE INTENTIONALLY LEFT BLANK

III. HIGH-FIDELITY MODELS FOR REUSABLE LAUNCH VEHICLES

A. INTRODUCTION

As stated in the motivation section of the introduction, model fidelity plays a key role in the accuracy of trajectory optimization solutions and consequently, has a direct influence on the vehicle's safety. This is especially true for complex vehicles such as the aging, soon to be retired, U.S. Space Shuttle and more modern RLV designs to include the X-33, X-34, X-37, X-40A, etc. There will probably always be controversy over whether or not using low-order models is sufficient or not for most of the engineering design and analysis of aerospace systems, but the bottom line is that is depends on the application. For the design and development of guidance and control (G&C) systems and methods, simplicity and robustness are indeed important. Since most G&C methods are based on simplified models, it begs the question, why? Why take unnecessary safety risks? Is it the complexity and challenge of modeling a high-fidelity model? Is it that no one knows how to solve the problem? Is it a matter of efficiency? Are the lower-fidelity models good enough? Although this work does not intend to answer these questions, it does provide some references to examples of why using lower-fidelity models may not be suitable for RLV G&C, address some issues with model fidelity, and provide results from solving a complex reentry problem using a 3-DOF model and a 6-DOF model.

The purpose of this chapter is three-fold. First, this chapter provides an overview description of the system under study – the X-33 RLV. Second, this chapter's main goal is to develop a 6-DOF model of a RLV and its environment for use in the high-fidelity trajectory optimization simulations presented in Chapter VI. This consists of developing the standard equations of motion (EoM) typically used in flight vehicle simulations and then a modified set of EoM more appropriate for trajectory optimization applications. Finally, a discussion of model fidelity with a homotopy path to modeling complex systems is presented.

B. DESCRIPTION OF THE SYSTEM UNDER STUDY

1. X-33 Reusable Launch Vehicle

a. Program

The X-33 program, started in 1996, was Lockheed Martin's advanced technology demonstrator for NASA's "next-generation" space launch vehicle program [160]. At the time, NASA was interested in lowering space access costs and sought solutions involving Single-Stage-to-Orbit (SSTO) concepts and prototypes. The X-33 half-scale prototype was to demonstrate its cutting-edge technologies (e.g., aerospike engines, load-bearing composite cryogenic fuel tanks, advanced thermal protection systems, etc.) that would lead to the development of the full-scale VentureStar RLV. The X-33 was expected to begin flight-test demonstrations in 1999 but, continued technology maturation problems led to program termination in 2001 [161]. Although the program was cancelled, NASA continued its Space Launch Initiative (SLI) with the 2nd Generation RLV Program starting from 2001 [162]. The following section summarizes the VentureStar's intended mission and the X-33's flight-test plan to provide a framework for the entry trajectory conditions used in this work.

b. Mission

The VentureStar was planned to take-off vertically, ascend to a Low Earth Orbit (LEO) carrying a small payload, perform minimal orbital activities, transition and enter the earth's atmosphere, descend through various intermediate flight segments (similar to the Space Shuttle Orbiter), and then ultimately land at a designated site. Unlike the VentureStar, the X-33 was intended only as a technology demonstrator and as such was to only ascend to a suborbital altitude of approximately 250,000 ft. Figure 3.1 illustrates the various mission phases with their corresponding sensor and control effector details.

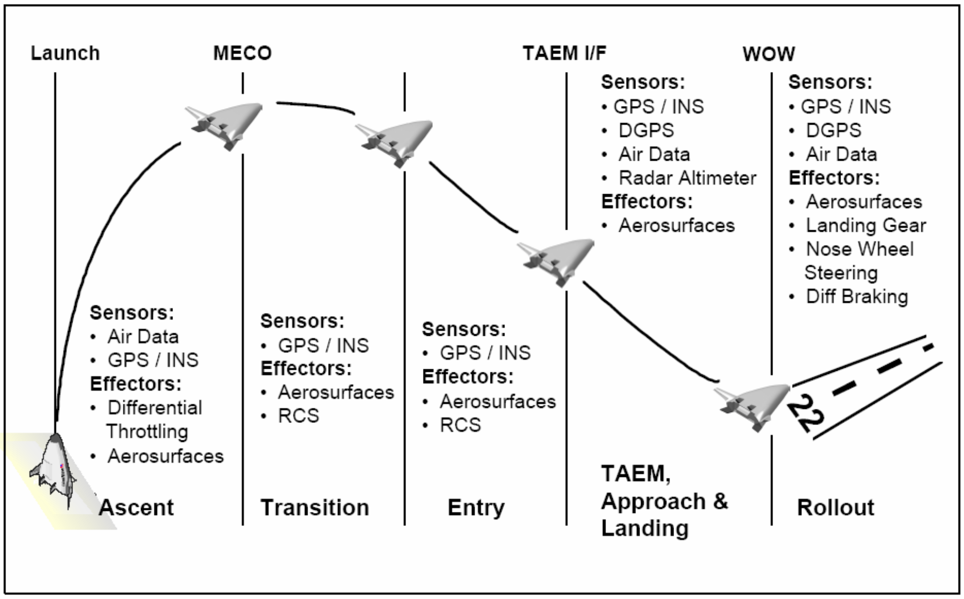


Figure 3.1 X-33 Flight Phases [163].

The mission profiles depicted in Figure 3.2, are representative of the three planned flight tests that were to originate at Edwards Air Force Base, CA. The landing sites included Silurian Dry Lake Bed (CA), Michael Army Air Field (UT), and Malmstrom Air Force Base (MT) at approximately 100 mi, 450 mi, and 950 mi (statute) downrange from Edwards, respectively.

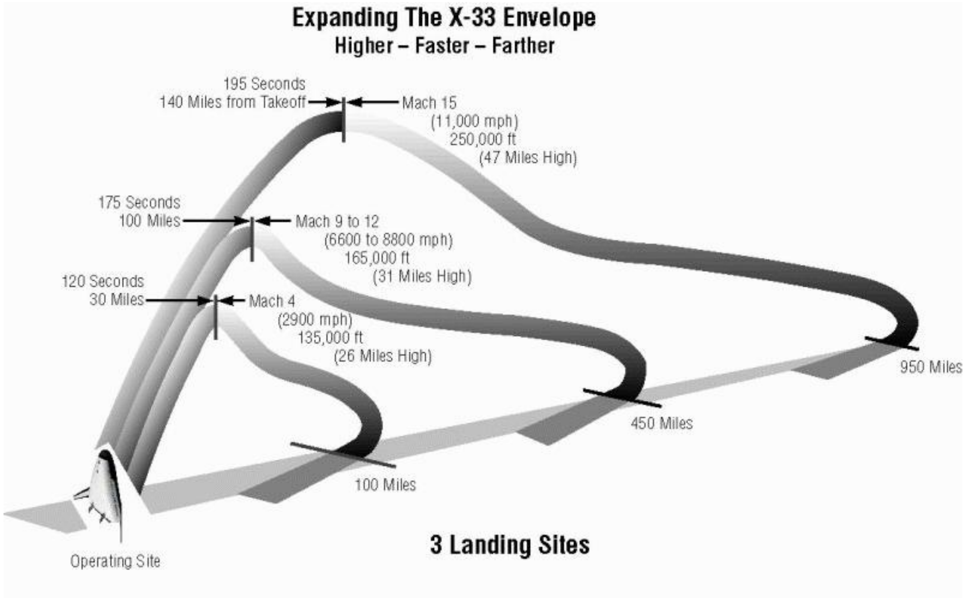


Figure 3.2 X-33 Flight-Test Mission Profiles [161].

Although this dissertation does not use actual orbital altitude as the initial altitude nor use reaction control system (RCS) for use in the upper-atmosphere where little to no dynamic pressure is available for effective aerosurfaces, the specific entry trajectory conditions were selected based on a combination of the X-33 flight-test mission profiles and those presented in Ref. [163]. Similar to Ref. [163], the entry conditions for this work assume a trajectory initiating from the International Space Station (ISS) with an orbital inclination of 51.6 deg with the final landing site as Kennedy Space Center (KSC), Florida. Depending on the exact timing of entry interface, the trajectories may have a small or large crossrange component – indicated by heading. In any case, the trajectories all approach KSC from the west descending over the Gulf of Mexico or south west over the Caribbean Sea. More specific information regarding the endpoint conditions for the various trajectory optimization problems will be provided in their respective chapters and sections.

c. Vehicle Characteristics

The X-33 is a lifting-body design with a L/D ratio ranging from approximately 1.25 to 4 for a Mach range from 0.10 to 30, respectively. Its gross lift-off weight and burn-out weight (after MECO) are approximately 280,000 lbs and 78,000 lbs, respectively. The vehicle's relative size compared to the full-scale VentureStar and the U.S. Space Shuttle is shown in Figure 3.3.

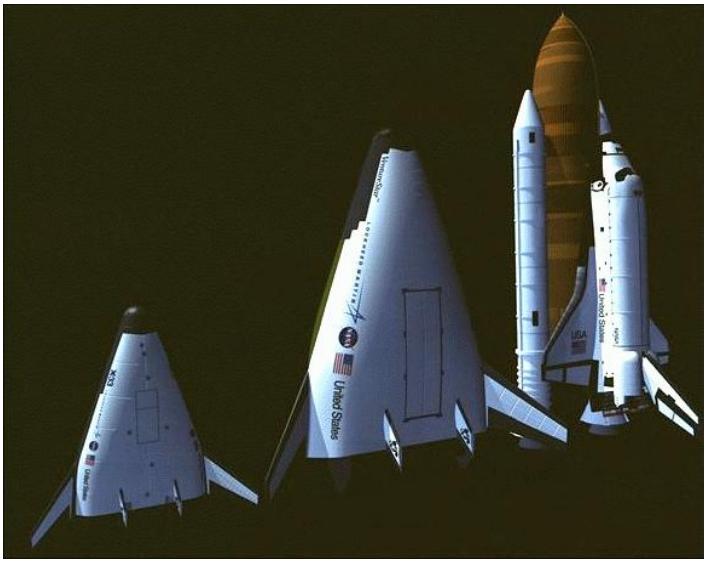


Figure 3.3 Size Comparison of X-33, VentureStar, and U.S. Space Shuttle [161].

The configuration and approximate dimensions are shown in Figure 3.4.

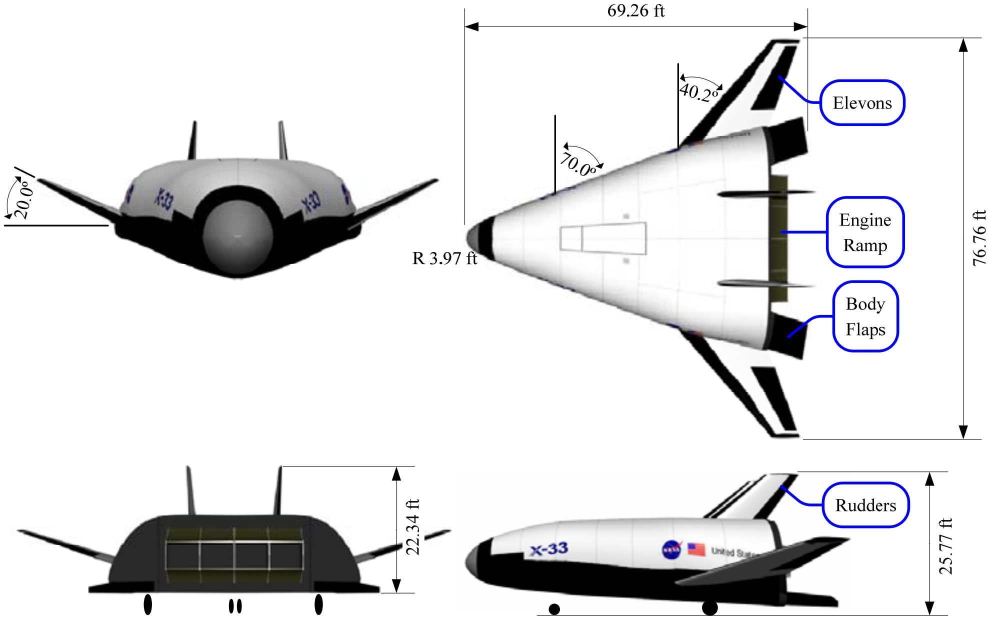


Figure 3.4 X-33 Configuration and Dimensions.

The X-33 has a unique control surface configuration. As shown in Figure 3.5, there is multiple redundancies provided by the in-board and out-board elevons, split body flaps, and dual rudders. It is common for launch vehicles to have redundant actuators, but to have redundant aerosurfaces provides the unique capability of reconfigurable controls. This allows compensation for changed aerodynamics and maintaining controllability as a result of control surface failures, vehicle degradation, aerodynamic uncertainties, etc. Implications of control surface failures and safe flight are discussed in Ref. [46]

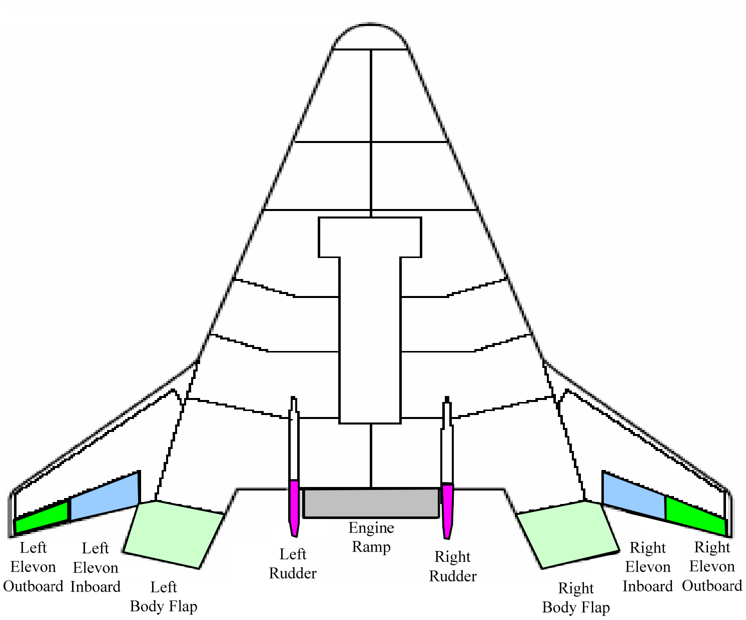


Figure 3.5 X-33 Redundant Aerosurface Configuration.

With the aerosurface configuration shown in Figure 3.5, the control surface deflection vector, also know as the effector vector, is represented by

$$\boldsymbol{\delta} = [\delta_{RE_{in}}, \delta_{RE_{out}}, \delta_{LE_{in}}, \delta_{LE_{out}}, \delta_{RF}, \delta_{LF}, \delta_{RR}, \delta_{LR}] \tag{3.1}$$

where $\delta_{RE_{in}}$ is the deflection position for the right elevon – inboard, $\delta_{RE_{out}}$ is for the right elevon – outboard, etc. A more thorough description of this effector vector is described in the aerodynamic model section (Sec. D.d). Additional vehicle parameters for the X-33 models used in this work are given in Table 3.1.

Table 3.1 X-33 Vehicle Parameters

Parameter	**Value**	**Unit**
Empty Mass, m	2455	$slugs$
Ref. Surface Area, S_{ref}	1608	ft^2
Ref. Aerodynamic Span, b	36.60	ft
Ref. Aerodynamic Chord, $\overline{c}$	63.20	ft
Ref. Aspect Ratio, AR	0.86	-
Ref. C.G. x-axis, CG_x	42	ft
Ref. C.G. y-axis, CG_y	0	ft
Ref. C.G. z-axis, CG_z	0.67	ft
Principal MOI x-axis, I_{xx}	434270	$slugs - ft^2$
Principal MOI y-axis, I_{yy}	961200	$slugs - ft^2$
Principal MOI z-axis, I_{zz}	1131541	$slugs - ft^2$
Cross-Product MOI, I_{xz}	17880	$slugs - ft^2$

2. Why the X-33?

As stated previously, the X-33 program was cancelled; however, considerable progress was made and extensive aerodynamic testing generated a high fidelity aerodynamic database for the vehicle that is quite useful for studies such as this. The selection of the vehicle to use for modeling and simulation is important for testing advanced G&C algorithms. For preliminary research or design work, it is common to use a simplified, generic vehicle model. However, for analyzing new G&C algorithms, it is essential that the vehicle model can adequately test the functionality of the proposed control techniques. In doing such, it is also useful to have previous test data available for feasibility studies. With this said, the X-33 RLV design makes an excellent test model for advanced G&C methods. Since the development of the X-33 was the forerunner for the 2nd generation RLV initiative in the early 90's, there exists a large amount of performance data and experimental aerodynamic data generated from wind-tunnel tests, proto-type hardware testing, computational fluid dynamic simulations, etc. In addition to the availability of data, the X-33's multiple redundant control surfaces make it an ideal test-bed for studying failure scenarios.

From a controls standpoint, the X-33 configuration has an added benefit for testing new methods. With eight aerodynamic control surfaces, consisting of right and left in-board elevons, right and left out-board elevons, right and left body flaps, and right and left rudders (see Figure 3.5), along with a reaction control system for thrust controlled maneuvers at high altitudes, the X-33 provides an excellent test model for exploring and simulating various control methodologies, especially for off-nominal conditions. Such conditions of interest are those that may require control reconfiguration, allocation, and adaptation such as the case of recovering from control surface failure(s), vehicle damage, or significant uncertainties and disturbances. Obviously, due to the abundance of data and its unique design configuration, the X-33 has been used extensively for research studies. These efforts have provided numerous reference trajectories that can be used for benchmark comparisons, etc.

C. PRELIMINARY FLIGHT MECHANICS

1. Reference Frames and Coordinate Systems

Since this work deals with flight in the sensible atmosphere, the reference frames and coordinate systems do not include some of those associated with space flight. However, there are various reference frames and coordinate systems available for modeling a RLV's position (i.e., geo-spatial location), orientation, and motion. Table 3.2 defines the various reference frames, classified into two primary systems: Earth system and body system, and the corresponding coordinate systems that define the Cartesian triad. Note that an orthonormal transformation will change one coordinate system to another.

Table 3.2 Reference Frames and Coordinate System Matrix (Adapted from [26])

	Reference Frames	**Coordinate Systems**
Earth System	**Inertial (Geocentric-Inertial)** F_I - Generic "fixed" frame, uniform rectilinear translation often relative to Earth's cm	ECI (Earth-centered inertial), origin at Earth's cm, axes in equatorial plane and along spin axis
	Earth-Centered F_E - "rigid" Earth	ECEF (Earth-centered, Earth-fixed), axes in equatorial plane and along spin axis
	Local-Tangent F_T - plane on Earth's surface $\perp$ line connecting Earth's cm and vehicle's cm	Tangent-plane, origin on Earth's surface
	Vehicle-Carried Vertical (Nav or Local Horizontal) F_V or F_N -translates with vehicle's cm	Origin at vehicle's cm, fixed geographic directions (NED)
Body System	**Body-Fixed** F_B - "rigid" vehicle	Origin at vehicle's cm, x aligned with body nose, z $\perp$ x (down), y $\perp$ XZ (right)
	Atmosphere-Fixed F_A - vehicle relative to local atmosphere	Origin at vehicle's cm, if atmosphere at rest → $F_A = F_E$
	Velocity / Wind F_W - relative wind	Origin at vehicle's cm, x aligned with V, z $\perp$ x (down), y $\perp$ XZ (right) -- $v, w = 0 \Rightarrow u = V$ & $\theta = \gamma$
	Stability F_S - for steady, symmetric flight $F_S = F_W$	Origin at vehicle's cm, x aligned with proj(V) onto body XZ plane, z $\perp$ x (down), y $\perp$ XZ (right)

An important reference frame is the body-fixed frame with a right-handed coordinate system originating at the vehicle's center of mass (cm).

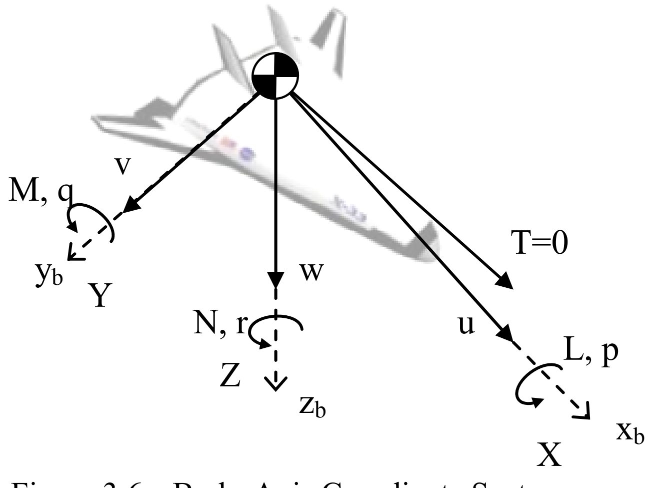


Figure 3.6 Body-Axis Coordinate System.

In accordance with Figure 3.6, the x-axis (longitudinal axis) extends the length of the body through the vehicle's nose, the y-axis (lateral axis) extends out the right wing, and the z-axis (vertical axis) points down. Table 3.3 defines the standard notation given by Figure 3.6 for the forces, moments, and velocities with respect to their corresponding body-axes.

Table 3.3 Standard Definitions of Forces, Moments, and Velocity Components (Adapted from [100])

	Roll (x-axis)	**Pitch (y-axis)**	**Yaw (z-axis)**
Angular Rates	p	q	r
Velocity Components	u	v	w
Aero. Force Components	X	Y	Z
Aero. Moment Components	L	M	N
MOI about each axis	Ix	Iy	Iz
Products of Inertia	Iyz	Ixz	Ixy

For velocity and wind-axes, the choice of using (u, v, w), (V, ξ, γ), or (V, α, β) depends on the specific application and/or desired analysis. Depending on the choice of an Earth Model, discussed in Sec. F.1, there is only one logical choice for the corresponding coordinate system. For example, the following illustrates typical earth-model-coordinate system maps:

1. Ellipsoidal-Earth Model → Geodetic Coordinates
2. Spherical-Earth Model → Geocentric Coordinates
3. Flat-Earth Model → Cartesian Coordinates

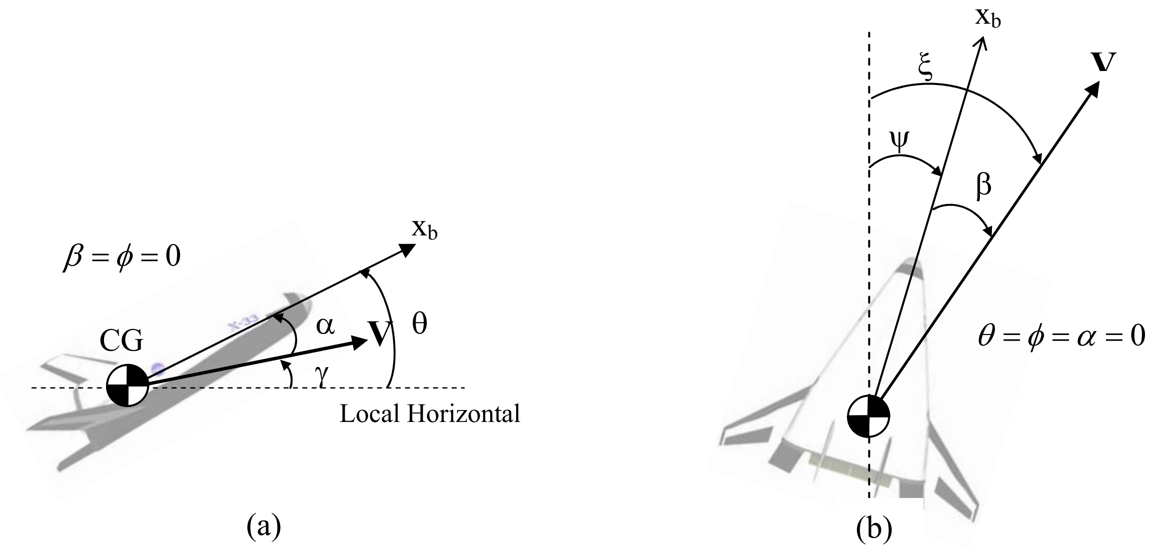
Before presenting the equations of motion, the commonly used flight angles must be defined along with their strongly coupled, often ignored, relationships.

2. Aircraft Flight Angles

Depending on the level of fidelity used in modeling flight vehicle motion, the defining angles may be represented as simple trigonometric relations such as for steady-state analysis or more complex spherical trigonometric relations such as for maneuvering flight. Reduced-order models and flight analysis often assume small angles such that some angles are considered insignificant and omitted from the development of the dynamic equations of motion. For example, some research assumes roll angle and bank angle are equivalent which as shown later is only valid for a small angle-of-attack (AoA). This can be a dangerous assumption leading to incorrect results especially where accurate solutions are needed. Another common assumption used in various research efforts are the following relationships between the "heading" angles (ξ, ψ, β) and the "pitch" angles (γ, θ, α)

$$\xi = \psi + \beta \tag{3.2}$$
$$\gamma = \theta - \alpha \tag{3.3}$$

These relationships are geometrically shown in Figure 3.7(a) and Figure 3.7(b).



(a)

(b)

Figure 3.7 Flight Angles: (a) Longitudinal Motion (b) Directional Motion.

As shown later, these relations are only valid for restricted maneuvers in which the vehicle remains in a "wings-level" orientation. Even though Kalviste pointed out in 1987 that these simple relations are being applied under conditions where they do not apply, it appears that these errors are still being made today [164]. For this reason and since it is important to consider the full, correct angle relationships when developing high-fidelity models for high-accuracy solutions, it is here that the full angle relations are re-developed.

Traditionally, stability and control analysis uses Euler and aerodynamic angles where as point-mass trajectory analysis uses flight-path angles. For trajectory optimization problems, it is beneficial to consider all angles since flight maneuverability is the primary factor driving such analysis. For example, the main objective of the reentry problem is to determine the vehicle's flight maneuvers (effected by control surface deflections) to get from one point (orbit or entry interface) to the other (landing).

a. Flight Path Angles

The flight path angles (ξ, γ, σ), geometrically shown in Figure 3.8, define the vehicle's trajectory through space usually given by the velocity vector relative to some local reference plane such as the so called "local horizontal" in the direction of North. In terms of a point-mass model, these angles characterize the *translational motion* of the vehicle. Note that the following nomenclature definitions are from Ref. [164].

ξ - flight path heading angle (i.e., "heading angle"), horizontal angle between a reference direction, usually North, and the projection of the velocity vector onto the local horizontal plane; positive rotation is typically from North to East

γ - flight path elevation angle, (i.e., "flight path angle"), vertical angle between the velocity vector and the local horizontal plane; positive rotation is up

σ - flight path bank angle, (i.e., "bank angle"), angle between the plane formed by the velocity vector and the lift vector and the vertical plane in the local horizontal frame that contains the velocity vector; positive rotation is clockwise about the velocity vector

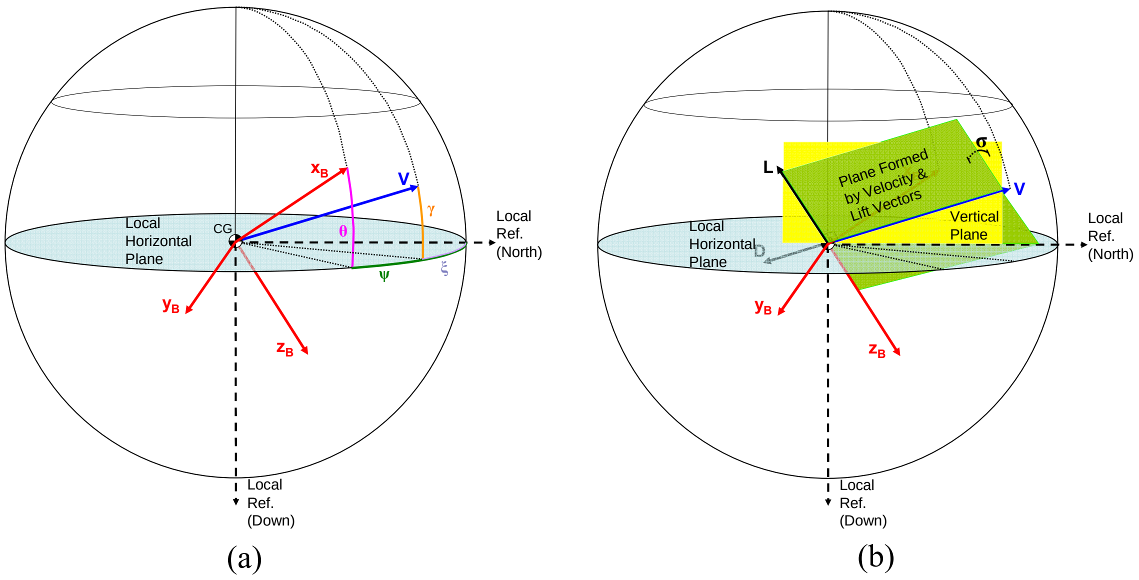


Figure 3.8 Flight Path Angles: (a) Elevation and Heading (b) Bank.

b. Aerodynamic Angles

The aerodynamic angles (α, β) are primary independent variables from which aerodynamic coefficients are defined, i.e., $C_x(\alpha, \beta, ...etc.)$. They basically define the direction of the velocity vector with respect to the vehicle's body-fixed axes. These angles are geometrically shown in Figure 3.9 and depend on the projection of the velocity vector onto the body x-z plane. Like the Euler angles as discussed next, the aerodynamic angles relate the body-frame with the wind-frame by a sequence of rotations. Note that the following nomenclature definitions are from Ref. [164].

α - angle-of-attack, angle between the body x-axis and the projection of the velocity vector onto the body x-z plane; positive rotation from z-axis towards x-axis

β - sideslip angle, angle between the velocity vector and the body x-z plane in body-axis coordinates; positive to the right of x-z plane

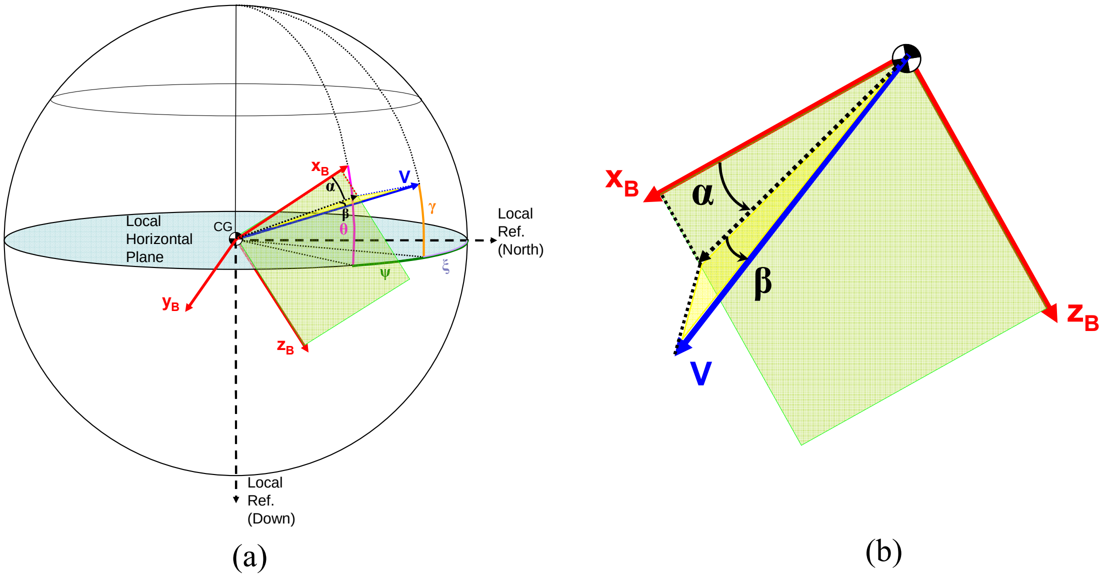


Figure 3.9 Aerodynamic Angles: (a) AoA and Sideslip Angle (b) Enlarged View from Opposite Side.

c. *Attitude Angles*

The attitude angles, usually given by the common Euler angles (ψ,θ,ϕ) geometrically illustrated in Figure 3.10, represent the vehicle's attitude orientation and characterize its *rotational motion.* In flight dynamics, this rotation is typically measured with respect to a vehicle-carried local horizontal frame and some other relevant reference system, in most cases the body-fixed system. Note that the following nomenclature definitions are from Ref. [164].

ψ - vehicle heading angle, (i.e., "yaw angle"), horizontal angle between reference direction, usually North, and the projection of the vehicle x-axis onto the local horizontal plane; positive rotation typically from North to East

θ - vehicle pitch angle, (i.e., "pitch angle"), vertical angle between vehicle x-axis and local horizontal plane; positive rotation is up

ϕ - vehicle roll angle, (i.e., "roll angle"), angle between vehicle x-z plane and local vertical plane; positive rotation is up

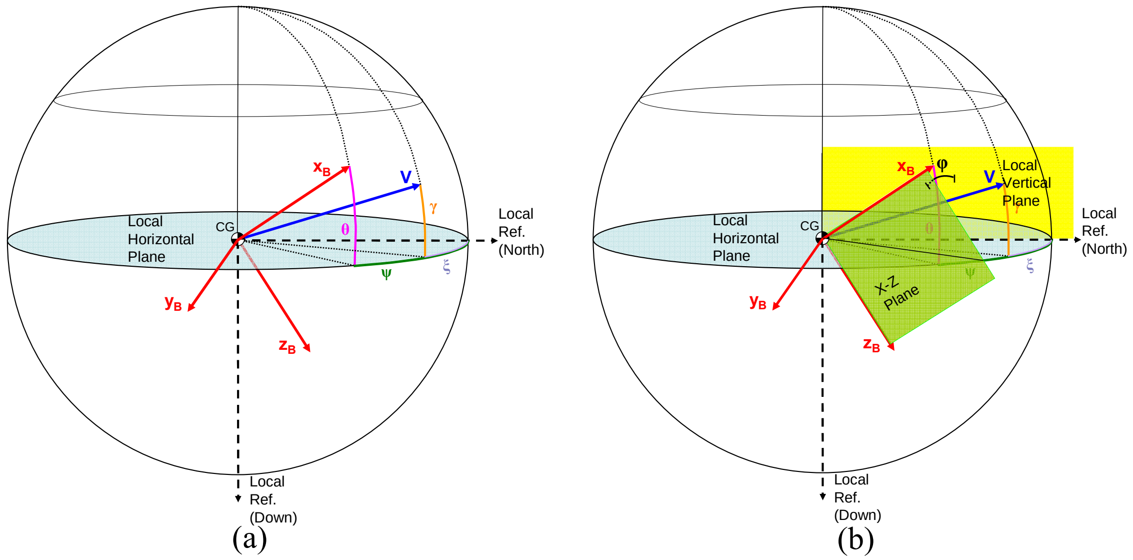


Figure 3.10 Euler Angles: (a) Pitch and Yaw (b) Roll.

d. Angular Relationships

For large-angle maneuvers, all the flight angles are needed to accurately define the vehicle's translational and rotational dynamics. There are essentially two approaches to deriving these angular relationships. The first approach consists of using trigonometry based on the physical geometry of the flight angles. This is done by projecting the angles onto a unit sphere as illustrated in Figure 3.8 to Figure 3.10 and then using spherical trigonometry (i.e., applying the laws of sine and cosine to spherical triangles) to derive their relationships. However, it is possible to achieve similar, but equally valid, relations without having to visualize these complex spherical geometries by performing a series of transformations between each of the coordinate systems via transformation matrix (TM) multiplications. This approach results in a Direction Cosine Matrix (DCM) and when set equal to the identity matrix it can provide equations that relate the various flight angles. The following discussion illustrates this second approach.

Inertial to Body-Fixed Frame TM $(F_V \rightarrow F_B)$

Consider the following Euler-angle rotation matrices,

$$\mathbf{R}_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix};\ \mathbf{R}_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix};\ \mathbf{R}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \quad (3.4)$$

The transformation from the inertial to the body-fixed reference frame (where the inertial frame, F_V, is taken as the vehicle-carried local horizontal) is achieved from the following matrix multiplication,

$$
\begin{gathered}
{}_V^B\mathbf{R} = \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\psi \\
\Rightarrow [T]^{BV} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \\
(3.5)
\end{gathered}
$$

<u>Inertial to Wind-Frame TM</u> $(F_V \rightarrow F_W)$

Similarly, consider the following flight-path-angle rotation matrices,

$$\mathbf{R}_\xi = \begin{bmatrix} \cos\xi & \sin\xi & 0 \\ -\sin\xi & \cos\xi & 0 \\ 0 & 0 & 1 \end{bmatrix};\ \mathbf{R}_\gamma = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix};\ \mathbf{R}_\sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\sigma & \sin\sigma \\ 0 & -\sin\sigma & \cos\sigma \end{bmatrix} \quad (3.6)$$

The transformation from the inertial to the wind reference frame is achieved from the following matrix multiplication,

$$
\begin{gathered}
{}_V^W\mathbf{R} = \mathbf{R}_\sigma \mathbf{R}_\gamma \mathbf{R}_\xi \\
\Rightarrow [T]^{WV} = \begin{bmatrix} \cos\gamma\cos\xi & \cos\gamma\sin\xi & -\sin\gamma \\ \sin\sigma\sin\gamma\cos\xi - \cos\sigma\sin\xi & \sin\sigma\sin\gamma\sin\xi + \cos\sigma\cos\xi & \sin\sigma\cos\gamma \\ \cos\sigma\sin\gamma\cos\xi + \sin\sigma\sin\xi & \cos\sigma\sin\gamma\sin\xi - \sin\sigma\cos\xi & \cos\sigma\cos\gamma \end{bmatrix} \\
(3.7)
\end{gathered}
$$

<u>Wind to Body-Frame TM</u> $(F_W \rightarrow F_B)$

Likewise, consider the following aero-angle rotation matrices,

$$\mathbf{R}_\beta = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix};\ \mathbf{R}_\alpha = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad (3.8)$$

The transformation from the wind frame to the body-fixed reference frame is achieved from the following matrix multiplication,

$$ {}^{B}_{W}\mathbf{R} = \mathbf{R}_{\alpha}\mathbf{R}_{\beta} \;\Rightarrow\; [T]^{BW} = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \tag{3.9} $$

Now, the combination of these rotation sequences leads to a DCM that satisfies

$$ [F_b] = [DCM]_{3\times3}[F_b] \tag{3.10} $$

where the DCM obviously must represent an identity matrix. This DCM is formed by multiplying the previous TMs

$$ [T]^{BB} = [T]^{BV}[T]^{VW}[T]^{WB} \Rightarrow [T]^{BV}\left[T^{VW}\right]^{T}\left[T^{WB}\right]^{T} \tag{3.11} $$

such that

$$ [DCM]_{3\times3} = [T]^{BV}[T]^{VW}[T]^{WB} = \begin{bmatrix} \mathrm{b11} & \mathrm{b12} & \mathrm{b13} \\ \mathrm{b21} & \mathrm{b22} & \mathrm{b23} \\ \mathrm{b31} & \mathrm{b32} & \mathrm{b33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.12} $$

This DCM relates all the flight angles and therefore any of the DCM elements can be used to solve for a desired angle. For example, isolating the sideslip angle term in the element b11 yields the following relation

$$ \begin{aligned} -\tan\alpha = {} & (\cos\theta\cos\psi\sin\sigma\sin\xi + \cos\theta\cos\psi\cos\sigma\sin\gamma\cos\xi - \cos\theta\sin\psi\sin\sigma\cos\xi \\ & + \cos\theta\sin\psi\cos\sigma\sin\gamma\sin\xi - \sin\theta\cos\sigma\cos\gamma) \,/\, (\cos\beta\cos\theta\cos\psi\cos\gamma\cos\xi \\ & + \cos\beta\cos\theta\sin\psi\cos\gamma\sin\xi + \cos\beta\sin\theta\sin\gamma + \sin\beta\cos\theta\cos\psi\cos\sigma\sin\xi \\ & - \sin\beta\cos\theta\cos\psi\sin\sigma\sin\gamma\cos\xi - \sin\beta\cos\theta\sin\psi\cos\sigma\cos\xi \\ & - \sin\beta\cos\theta\sin\psi\sin\sigma\sin\gamma\sin\xi + \sin\beta\sin\theta\sin\sigma\cos\gamma) \end{aligned} \tag{3.13} $$

$$ \begin{aligned} -\tan\beta = {} & (-\cos\theta\cos\psi\cos\sigma\sin\xi + \cos\theta\cos\psi\sin\sigma\sin\gamma\cos\xi + \cos\theta\sin\psi\cos\sigma\cos\xi \\ & + \cos\theta\sin\psi\sin\sigma\sin\gamma\sin\xi - \sin\theta\sin\sigma\cos\gamma) \,/\, (\cos\theta\cos\psi\cos\gamma\cos\xi \\ & + \cos\theta\sin\psi\cos\gamma\sin\xi + \sin\theta\sin\gamma) \end{aligned} \tag{3.14} $$

Although there is no obvious simplification of these equations, even with the aid of symbolic math tools like MAPLE, comparing them to relations derived from angles projected onto a unit sphere and applying spherical trigonometry reveals that Eqs. (3.13) and (3.14) are indeed valid.

Kalviste uses spherical trigonometry to derive the angles such that they are functions of each other [164].

$$\begin{aligned}(\gamma,\xi,\sigma) &= f(\alpha,\beta,\psi,\theta,\phi)\\ (\psi,\theta,\phi) &= f(\alpha,\beta,\gamma,\xi,\sigma)\\ (\alpha,\beta) &= f(\psi,\theta,\phi,\gamma,\xi)\end{aligned} \tag{3.15}$$

To verify the relations from the derived DCM (3.12), a comparison is made with the following Kalviste relations,

$$\begin{aligned}\sin\beta = \cos\gamma\left[\sin\theta\sin\phi\cos(\xi-\psi)+\cos\phi\sin(\xi-\psi)\right]\\ -\sin\gamma\cos\theta\sin\phi\end{aligned} \tag{3.16}$$

$$\begin{aligned}\sin\alpha\cos\beta = \cos\gamma\left[\sin\theta\cos\phi\cos(\xi-\psi)-\sin\phi\sin(\xi-\psi)\right]\\ -\sin\gamma\cos\theta\cos\phi\end{aligned} \tag{3.17}$$

-or-

$$\cos\alpha\cos\beta = \cos\gamma\left[\cos\theta\cos(\xi-\psi)\right]+\sin\gamma\sin\theta \tag{3.18}$$

Given $\gamma,\theta,\xi,$ and ψ, identities from Eq. (3.12) provide the identical α and β solutions as those from Eqs. (3.16) -(3.18).

In addition, it is now possible to confirm the over-simplified relations given by Eqs. (3.2) and (3.3) from the full flight-angle relationships, Eqs. (3.11) and (3.12). Assuming wings-level flight and zero sideslip, $\phi=\sigma=\beta=0$, and simultaneously solving Eq. (3.12), results in the limited relation for longitudinal motion in the vertical plane only,

$$\begin{aligned}&\theta = \tan^{-1}\left(\frac{\sin\alpha\cos\gamma+\cos\alpha\sin\gamma}{\cos\alpha\cos\gamma-\sin\alpha\sin\gamma}\right)\\ &\Rightarrow \tan\theta = \frac{\sin(\alpha+\gamma)}{\cos(\alpha+\gamma)} = \tan(\alpha+\gamma)\\ &\Rightarrow \theta = \alpha+\gamma\\ &\Rightarrow \gamma = \theta-\alpha\end{aligned} \tag{3.19}$$

Likewise, assuming $\theta=\phi=\sigma=\alpha=0$, and simultaneously solving Eq. (3.12), results in the limited relation for directional motion in the horizontal plane only,

$$\begin{aligned}
&\beta = -\sin^{-1}\left(\sin\psi\cos\xi - \cos\psi\sin\xi\right) \\
&\Rightarrow \sin\beta = -\sin(\psi - \xi) \\
&\Rightarrow \beta = -(\psi - \xi) \\
&\Rightarrow \xi = \psi + \beta
\end{aligned} \tag{3.20}$$

Another important clarification about the flight angle relationships has to do with rolling and banking. Kalviste concludes in perhaps the most complete document of the interrelationships among the angles, besides his Northrop Corp. report [165], that an aircraft does not roll about the velocity vector (stability x-axis) for coordinated maneuvers (zero side-slip), but rather an "axis of rotation" of the velocity vector [164]. This is due to the velocity vector changing directions in large-angle roll maneuvers. This is important because it is often assumed that roll angle and bank angle are the same. In fact, only during constant AoA and constant sideslip angles (i.e., constant velocity vector) is this true.

D. FULL-ORDER DYNAMICAL MODEL (6-DOF)

1. Standard Flight Vehicle Equations of Motion

The EoM of an aerospace vehicle are derived from rigid body dynamics that are based on Newton's second law. This section provides a rudimentary derivation of the standard EoM primarily used in flight vehicle applications. The definitions and equations presented herein are primarily adapted from Etkin's textbooks on flight dynamics, Refs. [99] and [166]. Additional details can be found in Refs. [98]-[100] and [166]-[168]. The following sections provide a brief introduction to a flight vehicle's position, velocity, acceleration, attitude, forces and moments, and how they come together to form position, velocity, body-rate, and Euler-angle equations.

a. Position, Velocity, and Acceleration

The position of the vehicle's center of gravity is typically with respect to some inertial frame of reference. For air vehicles in general, there are primary two coordinate systems used for modeling and simulation: earth-centered or local-tangent. For models using a spherical earth, spherical coordinates based on an earth-centered reference frame make the most sense. In this case, the vehicle's position is measured with respect to its distance from the center of the earth and its position over the earth's geographical latitude and longitude. For models that assume a flat-earth or are using a local tangent coordinate system, the position is measured in Cartesian x-y-z coordinates with respect to an arbitrary point over the earth's surface. For example, the x-distance would be a downrange distance corresponding to a geographical direction (e.g., East), the y-distance would be a crossrange distance corresponding to a geographical direction that is perpendicular to the x-distance (e.g., North), and the z-distance would be the altitude above the earth's surface. CAUTION: As shown in Figure 3.11, the body-fixed z-direction points down; however, altitude is measured up from the earth's surface; hence, in the negative z-direction.

The translation of the vehicle's position (cg) with respect to time is defined by the kinematical EoM. The following equations for the positional kinematics are given in spherical polar coordinates with respect to the vehicle-carried frame F_V (relative to Earth)

$$\begin{aligned} \dot{R} &= -v_D \\ \dot{\mu} &= \frac{1}{R\cos\lambda} v_E \\ \dot{\lambda} &= \frac{1}{R} v_N \end{aligned} \tag{3.21}$$

where $R = R_E + h$ is the geocentric radius, μ is the geocentric latitude, and λ is the geocentric longitude. Assuming zero winds $(\underline{W} = 0)$, the velocity components in the *vehicle-carried frame* can be written as

$$\begin{aligned} v_N &= V\cos\gamma\cos\xi \\ v_E &= V\cos\gamma\sin\xi \\ v_D &= -V\sin\gamma \end{aligned} \tag{3.22}$$

where V is the velocity of the vehicle's cg relative to Earth since $V^E = V^A + W \Rightarrow (V^E = V^A) \equiv V$, ξ is heading angle (measured positive clock-wise from the North), and γ is flight path angle (measured from the local horizontal). Note that the subscripts *N*, *E*, and *D* indicate that the reference frame's x,y,z-axes are aligned with the geographical North, East, Down directions. Substituting Eq. (3.22) into Eq. (3.21) yields the following spherical kinematical equations

$$\begin{aligned}\dot{R} &= V\sin\gamma \\ \dot{\mu} &= \frac{V}{R\cos\lambda}\cos\gamma\sin\xi \\ \dot{\lambda} &= \frac{V}{R}\cos\gamma\cos\xi\end{aligned} \tag{3.23}$$

For motion over a relatively small area of the Earth's surface, the Earth can be considered "locally flat" such that the vehicle-carried frame ("local horizontal") is parallel to the Earth-fixed frame, $F_V \,||\, F_E$ and $V_x \to \dot{x}_E$. This simplifies the kinematics to a Cartesian x,y,z-representation

$$\begin{aligned}\dot{x}_E &= V\cos\gamma\cos\xi \\ \dot{y}_E &= V\cos\gamma\sin\xi \\ \dot{z}_E &= -V\sin\gamma\end{aligned} \tag{3.24}$$

An alternative system of equations is expressed with velocity components in the *body-fixed frame,* $V_b = [u_b, v_b, w_b]^T$,

$$\begin{bmatrix}\dot{\lambda}R \\ \dot{\mu}R\cos\lambda \\ -\dot{R}\end{bmatrix} = [\mathbf{T}]^{BV}\begin{bmatrix}u_b \\ v_b \\ w_b\end{bmatrix} \tag{3.25}$$

where the linear velocity components, as shown in Figure 3.6 and Figure 3.11, are

$$\begin{aligned}u_B &= V\cos\alpha\cos\beta \\ v_B &= V\sin\beta \\ w_B &= V\sin\alpha\cos\beta\end{aligned} \tag{3.26}$$

and $[\mathbf{T}]^{BV}$ is the transformation matrix from the inertial earth-fixed frame to the body-fixed frame $(F_V \to F_B)$ given as

$$[\mathbf{T}]^{BV} = \begin{bmatrix}\cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta\end{bmatrix} \tag{3.27}$$

Substituting the appropriate equations into Eq. (3.25) and comparing to the kinematics in Eq. (3.24) results in

$$\begin{bmatrix} \dot{\lambda} r \\ \dot{\mu} r \cos\lambda \\ -\dot{r} \end{bmatrix} = \begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{z}_E \end{bmatrix} \tag{3.28}$$

Or, it is more common to find the right-hand-side (RHS) of Eq. (3.25) expanded to form the position equations, such that

$$\begin{aligned} \dot{x}_E &= u(\cos\theta\cos\psi) + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) \\ &\qquad + w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\[1em] \dot{y}_E &= u(\cos\theta\sin\psi) + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) \\ &\qquad + w(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\[1em] \dot{z}_E &= u(-\sin\theta) + v(\sin\phi\cos\theta) + w(\cos\phi\cos\theta) \end{aligned} \tag{3.29}$$

Note that in the presence of wind, described by the wind vector $\underline{W} = [W_x, W_y, W_z]^T$ the velocity terms can be written as

$$\begin{aligned} u &= u_B + W_x \\ v &= v_B + W_y \\ w &= w_B + W_z \end{aligned} \tag{3.30}$$

CAUTION: All the following velocity terms assume zero wind (i.e., $\underline{W} = 0$) such that $u = u_B$. If using spherical velocity terms, the linear velocities in Eq. (3.26) can be substituted into Eq. (3.29) to give

$$\begin{aligned} \dot{x}_E &= V\cos\alpha\cos\beta(\cos\theta\cos\psi) + V\sin\beta(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) \\ &\qquad + V\sin\alpha\cos\beta(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\[1em] \dot{y}_E &= V\cos\alpha\cos\beta(\cos\theta\sin\psi) + V\sin\beta(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) \\ &\qquad + V\sin\alpha\cos\beta(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\[1em] \dot{z}_E &= V\cos\alpha\cos\beta(-\sin\theta) + V\sin\beta(\sin\phi\cos\theta) + V\sin\alpha\cos\beta(\cos\phi\cos\theta) \end{aligned} \tag{3.31}$$

The $\dot{x}_E, \dot{y}_E, \dot{z}_E$-equations are resolved in the body-axes frame (F_B), but when in the wind-axes frame (F_B), the x-axis is aligned with velocity vector, $\underline{V}$ such that

$v = w = 0$, $u = |\underline{V}|$, $\theta = \gamma$, and $\psi = \xi$. Substituting these into Eq. (3.29) gives Eq. (3.24).

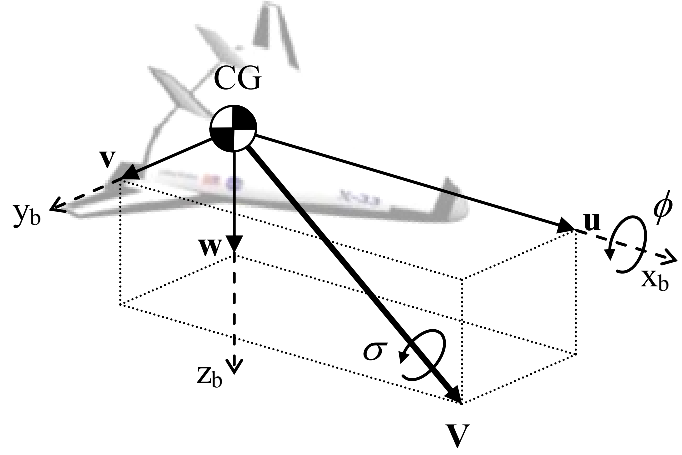


Figure 3.11 Velocity Components as Projections of Velocity Vector

It is clear from Figure 3.11 that the linear velocity terms introduced in Eq. (3.26) are simply projections of the total airspeed, **V**. Therefore, this speed is defined as the velocity magnitude given by

$$V = \sqrt{u^2 + v^2 + w^2} \tag{3.32}$$

In addition, the aerodynamic wind-relative angles, α and β, are determined from the linear velocity terms

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \tag{3.33}$$

$$\beta = \tan^{-1}\left(\frac{u}{V}\right) = \sin^{-1}\left(\frac{v}{V}\right) \tag{3.34}$$

Instead of using Cartesian velocity vector components (u, v, w) without wind, using the spherical velocity terms (V, α, β) makes the problem physically intuitive and easier to visualize the appropriate constraints. This has little, if any, numerical effects since these coordinate systems are directly related by Eqs. (3.32)-(3.34). Now, the velocity equations are found from directly differentiating Eqs. (3.32), (3.33), and (3.34) to get

$$\dot{V} = \frac{1}{V}(u\dot{u} + v\dot{v} + w\dot{w}) \tag{3.35}$$

$$\dot{\alpha} = \frac{u\dot{w} - w\dot{u}}{u^2 + w^2} \tag{3.36}$$

$$\dot{\beta} = \frac{V\dot{v} - v\dot{V}}{V^2\sqrt{1 - v^2 / V^2}} \tag{3.37}$$

respectively. Also, since these equations will be applied to a gliding reentry problem (a.k.a. "dead-stick"), the thrust terms are omitted. If thrust was available, the following terms could be added such that

$$\dot{V} = ... + \left[\frac{T}{m}\cos\alpha\cos\beta\right] \tag{3.38}$$

$$\dot{\alpha} = ... - \left[\frac{T\sin\alpha}{mV\cos\beta}\right] \tag{3.39}$$

$$\dot{\beta} = ... + \left[\frac{T\sin\beta\cos\alpha}{mV}\right] \tag{3.40}$$

Likewise, if using the vehicle-carried reference frame, the velocity magnitude is represented by the components given in Eq. (3.22),

$$V = \sqrt{{v_N}^2 + {v_E}^2 + {v_D}^2} \tag{3.41}$$

with the velocity-relative angles, γ and ξ, determined as

$$\gamma = \sin^{-1}\left(\frac{-v_D}{V}\right) \tag{3.42}$$

$$\xi = \cos^{-1}\left(v_N / \sqrt{{v_N}^2 + {v_E}^2}\right) = \sin^{-1}\left(v_E / \sqrt{{v_N}^2 + {v_E}^2}\right) \tag{3.43}$$

Now that position, velocity, and acceleration are defined, the next section introduces the forces and moments required to formulate the complete translational EoM and introduce the rotational EoM. For example, the current form of Eqs. (3.35)-(3.37) need to be modified to include force terms.

b. Forces and Moments

The total force generated on a vehicle during atmospheric flight primarily consists of contributions from aerodynamics, gravity, and propulsion (a.k.a thrust) represented as

$$F_{total} = F_{aero} + F_{grav} + F_{prop} \tag{3.44}$$

Assuming there is no wind contribution in the linear velocity terms (i.e., $\underline{W} = 0$), the aerodynamic forces along the corresponding body-axes (x_B, y_B, z_B) and that depend on the current flight condition are represented as

$$\begin{aligned} F_{x(aero)} &= X = \bar{q} S_{ref} C_x \\ F_{y(aero)} &= Y = \bar{q} S_{ref} C_y \\ F_{z(aero)} &= Z = \bar{q} S_{ref} C_z \end{aligned} \tag{3.45}$$

where $\bar{q}$ is the dynamic pressure given as $\bar{q} = \frac{1}{2}\rho(z)V^2$, S_{ref} is the reference area, and C_X, C_Y, C_Z are the corresponding aerodynamic coefficients that are functions of various variables depending on the coefficient of interest as discussed later. The local gravity terms for each axis are given by

$$\begin{aligned} g_x &= -g\sin\theta \\ g_y &= g\sin\phi\cos\theta \\ g_z &= g\cos\phi\cos\theta \end{aligned} \tag{3.46}$$

such that the forces due to gravity are

$$\begin{aligned} F_{x(grav)} &= mg_x = -mg\sin\theta \\ F_{y(grav)} &= mg_y = mg\sin\phi\cos\theta \\ F_{z(grav)} &= mg_z = mg\cos\phi\cos\theta \end{aligned} \tag{3.47}$$

Omitting the propulsion terms (un-powered reentry) and combining Eqs. (3.45)-(3.47), the external forces are written as

$$\begin{aligned} F_x &= X + mg_x \\ F_y &= Y + mg_y \\ F_z &= Z + mg_z \end{aligned} \tag{3.48}$$

Also, the external force can be represented as the time derivative of linear momentum

$$F = \frac{d(mV)}{dt} \Rightarrow \frac{\delta(mV)}{\delta t} + \omega \times (mV) \tag{3.49}$$

where $\omega = [p,q,r]^T$ is the angular velocity vector and $V = [u,v,w]^T$ is the velocity vector. Note that the angular velocity consists of the local body-frame velocity and the Earth's velocity relative to the vehicle. For the reentry problem, it is assumed that mass is constant since there is usually no fuel available for propulsion. Hence, the forces become

$$\begin{aligned} F_x &= m[\dot{u}+(q_E+q)w-(r_E+r)v] \\ F_y &= m[\dot{v}+(r_E+r)u-(p_E+p)w] \\ F_z &= m[\dot{w}+(p_E+p)v-(q_E+q)w] \end{aligned} \tag{3.50}$$

For the flat-earth approximation or assuming negligible earth rotation, $\omega_E=[p_E,q_E,r_E]^T=\underline{0}$; hence, Eq. (3.50) reduces to

$$\begin{aligned} F_x &= m(\dot{u}+qw-rv) \\ F_y &= m(\dot{v}+ru-pw) \\ F_z &= m(\dot{w}+pv-qw) \end{aligned} \tag{3.51}$$

Equating Eqs. (3.51) with (3.48) yields

$$\begin{aligned} X &= m(\dot{u}+qw-rv+g\sin\theta) \\ Y &= m(\dot{v}+ru-pw-g\cos\theta\sin\phi) \\ Z &= m(\dot{w}+pv-qu-g\cos\theta\cos\phi) \end{aligned} \tag{3.52}$$

Equation (3.52) constitutes the body-axes, force equations for a vehicle assuming a flat-earth with no wind. The velocity equations given by Eqs. (3.35)-(3.37) can now be converted to a more useful form by solving for $(\dot{u},\dot{v},\dot{w})$ in Eq. (3.52),

$$\begin{aligned} \dot{u} &= \frac{X}{m}+g\sin\theta+rv-qw \\ \dot{v} &= \frac{Y}{m}-g\cos\theta\sin\phi-ru+pw \\ \dot{w} &= \frac{Z}{m}-g\cos\theta\cos\phi+qu-pv \end{aligned} \tag{3.53}$$

and substituting these into Eqs. (3.35)-(3.37) yields

$$\begin{aligned} \dot{V} &= \left[\frac{X}{m}-g\sin\theta\right]\cos\alpha\cos\beta+\left[\frac{Y}{m}+g\sin\phi\cos\theta\right]\sin\beta \\ &\qquad +\left[\frac{Z}{m}+g\cos\phi\cos\theta\right]\sin\alpha\cos\beta \\ \dot{\alpha} &= \frac{1}{mV\cos\beta}\left[Z\cos\alpha-X\sin\alpha\right]+\frac{g}{V\cos\beta}\left[\cos\phi\cos\theta\cos\alpha+\sin\theta\sin\alpha\right] \\ &\qquad +q-\tan\beta\left[p\cos\alpha+r\sin\alpha\right] \qquad (3.54) \\ \dot{\beta} &= \frac{\cos\beta}{mV}\left[Y+mg\sin\phi\cos\theta\right]+p\sin\alpha-r\cos\alpha-\frac{\sin\beta}{mV}\left[Z\sin\alpha+X\cos\alpha\right] \\ &\qquad -\frac{g\sin\beta}{V}\left[\cos\phi\cos\theta\sin\alpha-\sin\theta\cos\alpha\right] \end{aligned}$$

where the aerodynamic forces and moments are

$$
\begin{aligned}
X &= \frac{1}{2}\rho(z)V^2 S_{ref} C_X(\alpha,\beta,M,\underline{\delta}) && \text{Axial Force} \\
Y &= \frac{1}{2}\rho(z)V^2 S_{ref} C_Y(\beta,M,\underline{\delta}) && \text{Side Force} \\
Z &= \frac{1}{2}\rho(z)V^2 S_{ref} C_Z(\alpha,\beta,M,\underline{\delta}) && \text{Normal Force} \\
L &= \frac{1}{2}\rho(z)V^2 S_{ref} b C_l(\beta,M,\underline{\delta}) && \text{Rolling Moment} \\
M &= \frac{1}{2}\rho(z)V^2 S_{ref} \overline{c} C_m(\alpha,\beta,M,\underline{\delta}) && \text{Pitching Moment} \\
N &= \frac{1}{2}\rho(z)V^2 S_{ref} b C_n(\beta,M,\underline{\delta}) && \text{Yawing Moment}
\end{aligned}
\tag{3.55}
$$

Equation (3.54) combined with Eq. (3.55) constitutes the body-axes, velocity equations for a vehicle assuming a flat-earth with no wind. *NOTE:* The aerodynamic coefficients $(C_X, C_Y, C_Z, C_l, C_m, C_n)$ are further explained in subsection d to follow.

Since most aerospace vehicles are not extremely long with irregular mass distributions, it is assumed that the gravity field has negligible effects on the moments. Therefore, the external moments acting on the vehicle are primarily caused by aerodynamic forces and can be represented as

$$
\begin{aligned}
M_{x(aero)} &= L = \overline{q} S_{ref} b C_l \\
M_{y(aero)} &= M = \overline{q} S_{ref} \overline{c} C_m \\
M_{z(aero)} &= N = \overline{q} S_{ref} b C_n
\end{aligned}
\tag{3.56}
$$

Also, the moments on the vehicle can be represented as the time derivative of angular momentum

$$
M = \frac{d(h)}{dt} \Rightarrow \frac{\delta(h)}{\delta t} + \omega \times h \tag{3.57}
$$

where *h* is the "moment of momentum" and depends on the vehicle's moment of inertia matrix such that

$$
h = I\omega = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.58}
$$

Assuming an xz-plane of symmetry, as for most flight vehicles, the moments are given as

$$\begin{aligned} L &= I_{xx}\dot{p} - I_{xz}(\dot{r} + pq) + (I_{zz} - I_{yy})qr \\ M &= I_{yy}\dot{q} + I_{xz}(p^2 - r^2) + (I_{xx} - I_{zz})rp \\ N &= I_{zz}\dot{r} - I_{xz}(\dot{p} - qr) + (I_{yy} - I_{xx})pq \end{aligned} \tag{3.59}$$

Note that this can further be simplified if the axes are principal, i.e., $I_{xz} = 0$. Solving Eq. (3.59) for the angular accelerations yields,

$$\begin{aligned} \dot{p} &= \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}^{\ 2}}\left[L - \left(I_{zz} - I_{yy}\right)qr + I_{xz}qp\right] + \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^{\ 2}}\left[N + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right] \\ \dot{q} &= \frac{1}{I_{yy}}\left[M - \left(I_{xx} - I_{zz}\right)pr - I_{xz}\left(p^2 - r^2\right)\right] \\ \dot{r} &= \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^{\ 2}}\left[L + \left(I_{yy} - I_{zz}\right)qr + I_{xz}qp\right] + \frac{I_{x}}{I_{xx}I_{zz} - I_{xz}^{\ 2}}\left[N + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right] \end{aligned} \tag{3.60}$$

Equation (3.60) constitutes the body-axes moment equations for a vehicle assuming a flat-earth with no wind.

c. Attitude Orientation

The vehicle's orientation in terms of body attitude can be described by Euler angles, Direction Cosines, or quaternions.

(1) Euler Angles. Euler angles, as defined in Sec. C.2.c, are generated by a series of rotations about each independent axis. Repeating Eq. (3.4) here for consistency, the individual rotation matrices are given by

$$\mathbf{R}_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix};\ \mathbf{R}_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix};\ \mathbf{R}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{3.61}$$

Note that Euler angles are not uniquely defined since the sequence of rotations changes the resulting transformation matrix. These Euler-rotation matrices are used to express the angular velocities about the body axes. Since the coordinate system for the Euler angle rates is not orthogonal, each rate must be transformed to the aircraft's corresponding body axis by another rotation. This series of transformations can be succinctly written as

$$\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\psi \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \mathbf{R}_\phi \mathbf{R}_\theta \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_\phi \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \tag{3.62}$$

$$\Rightarrow \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\phi \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.63}$$

Now, the rates are determined by inverting Eq. (3.63),

$$\begin{aligned} \dot{\phi} &= p + \left(q\sin\phi + r\cos\phi\right)\tan\theta \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= \left(q\sin\phi + r\cos\phi\right)\sec\theta \end{aligned} \tag{3.64}$$

Equation (3.64) constitutes the body-rate equations for a vehicle assuming a flat-earth with no wind. Although useful body-fixed relationships, it is important to note the limitation of pitch angle in these rate equations. For $\theta = \pm 90^\circ$, the roll rate $\dot{\phi}$ and yaw rate $\dot{\psi}$ are undefined (i.e., singular).

(2) Direction Cosine Matrix (DCM). The DCM was originally the predominant method of representing attitude in early strapdown and inertial navigation systems. Briefly introduced in Sec. C.2.d, the DCM is essentially a coordinate transformation matrix formed by a series of vector dot products (i.e., direction cosines). For example, consider two vectors, $\hat{a}_i$ and $\hat{b}_j$, such that when taking their dot product yields a nine-parameter DCM that rotates from one reference frame (*B*) to another (*A*)

$$[C_{ij}]^{AB} := \begin{bmatrix} \hat{a}_1 \cdot \hat{b}_1 & \hat{a}_1 \cdot \hat{b}_2 & \hat{a}_1 \cdot \hat{b}_3 \\ \hat{a}_2 \cdot \hat{b}_1 & \hat{a}_2 \cdot \hat{b}_2 & \hat{a}_2 \cdot \hat{b}_3 \\ \hat{a}_3 \cdot \hat{b}_1 & \hat{a}_3 \cdot \hat{b}_2 & \hat{a}_3 \cdot \hat{b}_3 \end{bmatrix} := \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{3.65}$$

where each element represents a cosine of the angle between the i-th axis of the A-frame and the j-th axis of the B-frame; hence,

$$\begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{bmatrix} = C^{AB} \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} \tag{3.66}$$

Due to limited computational efficiency from the redundancies in the nine-parameters and the required orthogonalization, this approach to attitude representation is not widely used anymore [170].

(3) Quaternions. Originally invented by Sir William Rowan Hamilton around 1843 [171] , their use in attitude dynamic simulations was not popular until the early 1960's [170]. Quaternions are of interest since their implementation in the EoM will avoid singularities such as those encountered when using Euler angles.

As previously stated and repeated here for clarity, the orthonormal transformation matrix (inertial Earth-frame to body-frame) with Euler rotation sequence: yaw, pitch, roll (ψ,θ,ϕ) is given by

$$[T]^{BE}=\begin{bmatrix}\cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta\\ (-\cos\phi\sin\psi+\sin\phi\sin\theta\cos\psi) & (\cos\phi\cos\psi+\sin\phi\sin\theta\sin\psi) & \sin\phi\cos\theta\\ (\sin\phi\sin\psi+\cos\phi\sin\theta\cos\psi) & (-\sin\phi\cos\psi+\cos\phi\sin\theta\sin\psi) & \cos\phi\cos\theta\end{bmatrix} \quad (3.67)$$

Quaternions define the orientation of one axis system relative to another by characterizing a single rotation about one axis, called the Euler Axis. Quaternions have four components, $\underset{=}{Q}=\left[q_0,q_1,q_2,q_3\right]^T$, where q_1, q_2, q_3 define a unit vector for the Euler Axis and q_0 is related to the magnitude of the angle of rotation about the Euler Axis. The quaternion rates are represented by the following (4x4)-skew-symmetric matrix of angular rates

$$\begin{Bmatrix}\dot{q}_0\\ \dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{Bmatrix}=\frac{1}{2}\begin{bmatrix}0 & -r & -q & -p\\ r & 0 & -p & q\\ q & p & 0 & -r\\ p & -q & r & 0\end{bmatrix}\begin{Bmatrix}q_0\\ q_1\\ q_2\\ q_3\end{Bmatrix} \quad (3.68)$$

and each quaternion can be expanded in terms of Euler angles such that

$$\begin{aligned}q_0&=\cos(\psi/2)\cos(\theta/2)\cos(\phi/2)+\sin(\psi/2)\sin(\theta/2)\sin(\phi/2)\\ q_1&=\sin(\psi/2)\cos(\theta/2)\cos(\phi/2)-\cos(\psi/2)\sin(\theta/2)\sin(\phi/2)\\ q_2&=\cos(\psi/2)\sin(\theta/2)\cos(\phi/2)+\sin(\psi/2)\cos(\theta/2)\sin(\phi/2)\\ q_3&=\cos(\psi/2)\cos(\theta/2)\sin(\phi/2)-\sin(\psi/2)\sin(\theta/2)\cos(\phi/2)\end{aligned} \quad (3.69)$$

If needed as an output, the Euler angles can now be determined by solving the system given in Eq. (3.69) to yield

$$\begin{aligned}
\tan\psi &= \frac{2(q_0 q_1 + q_2 q_3)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \\
\sin\theta &= -2(q_1 q_3 - q_0 q_2) \\
\tan\phi &= \frac{2(q_1 q_2 + q_0 q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}
\end{aligned} \tag{3.70}$$

A useful property of quaternions is the orthogonality constraint given by

$$q_0{}^2 + q_1{}^2 + q_2{}^2 + q_3{}^2 = 1 \tag{3.71}$$

and the orthonormal transformation matrix, originally given by Eq. (3.67), is now re-written as

$$[T]^{BE} = \begin{bmatrix} q_0^2 - q_1^2 - q_2^2 + q_3^2 & 2(q_0 q_1 + q_2 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) \\ 2(q_0 q_2 + q_1 q_3) & 2(q_1 q_2 - q_0 q_3) & q_0^2 + q_1^2 - q_2^2 - q_3^2 \end{bmatrix} \tag{3.72}$$

(4) Tradeoffs between Euler Angles, DCMs, and Quaternions. Overall, the use of a particular set of attitude equations depends on the intended application. Comparing the structure of the equations, the quaternion representation consists of four linear differential equations as opposed to nine (or six) linear equations for direction cosines or three nonlinear trigonometric equations for Euler angles. The primary advantages of using quaternions for online attitude calculations are that they are computationally simple and "globally" nonsingular. On the other hand, they are less intuitive and more difficult to implement. Plus, Euler angles are required if coordinate systems do not coincide at initial time and since they are not readily available; there is the burden of additional calculations. In other words, initialization of quaternions requires the calculation of Euler angles anyway. Also, when using quaternions with estimation techniques such as in Kalman Filtering, it is possible that a sigularitiy in the covariance matrix may result from the lack of independence of the four quaternion components [170]. This leads to an un-observable condition when implementing on real hardware. The following chart, Table 3.4, summarizes the advantages and disadvantages of the three approaches to representing attitude orientation.

Table 3.4 Comparison of Three Methods for Attitude Simulation Equations

	Euler Angles	DCM	Quaternions
Advantage	1. Only 3 equations 2. Direct Euler angles 3. Direct initialization	1. 6 linear equations 2. Direct TM 3. No singularities	1. 4 linear equations 2. Orthog. Condition 3. Comp. efficient 4. No singularities*
Disadvantage	1. Nonlinear equations 2. Singularities 3. Indirect TM	1. Indirect Euler angles 2. Indirect initialization 3. Comp. inefficient	1. Indirect Euler Angles 2. Indirect initialization 3. Indirect TM

* *Exception – covariance matrix singularity possible in estimation applications*

d. Aerodynamic Model

(1) Aerodynamic Forces and Moments. Of the forces given by Eq. (3.44), the aerodynamic forces (3.45) are by far the most complex, depending on a large number of variables. The aerodynamic forces and moments have various functional dependencies that can significantly influence simulations based on the modeling techniques employed. These nonlinear dependencies can create measurement and modeling problems that ultimately effect computer simulations. In general, the aerodynamic forces and moments can be represented as a function of flow characteristics such as Mach and Reynolds numbers (*M*, *Re*), aerodynamic angles and rates (α, β, $\dot{\alpha}$, $\dot{\beta}$), body angular rates (*p, q, r*), control surface deflections (δ_a, δ_e, δ_r), center of gravity (*cg*), power in the form of thrust, aerodynamic shape, scale, etc. [97]. The majority of the dependencies come from the wind-relative incidence angles, body rates, and control surface deflections that influence the non-dimensional aerodynamic coefficients.

The aerodynamic data used in this work is explicit functions of α, β, M, and $\underline{\delta}$ such that

$$C_A, C_Y, C_N, C_l, C_m, C_n = f(\alpha, \beta, M, \underline{\delta}) \tag{3.73}$$

Or more specifically,

$$\begin{aligned} C_A, C_m, C_N &= f(\alpha, \beta, M, \underline{\delta}) \\ C_l, C_Y, C_n &= f(\beta, M, \underline{\delta}) \end{aligned} \tag{3.74}$$

where $\underline{\delta}$ is the control surface deflection vector, commonly called the control "effector vector", and for this problem is given by

$$\underline{\delta} = [\delta_{RE_{in}},\ \delta_{RE_{out}},\ \delta_{LE_{in}},\ \delta_{LE_{out}},\ \delta_{RF},\ \delta_{LF},\ \delta_{RR},\ \delta_{LR}] \tag{3.75}$$

such that

$$\left\{ \underline{\delta} = \underline{u} \in \mathbb{R}^8 : \underline{u}^L \le \underline{u} \le \underline{u}^U \right\} \tag{3.76}$$

The aero-coefficients listed in Eqs. (3.73) and (3.74) correspond to the force and moment definitions given in Eq. (3.55) and are defined as follows:

$$\begin{aligned} C_A &\triangleq \text{Axial Force Coefficient along Body x-axis} \\ C_Y &\triangleq \text{Side Force Coefficient along Body y-axis} \\ C_N &\triangleq \text{Normal Force Coefficient along Body z-axis} \\ C_l &\triangleq \text{Rolling Moment Coefficient about Body x-axis} \\ C_m &\triangleq \text{Pitching Moment Coefficient about Body y-axis} \\ C_n &\triangleq \text{Yawing Moment Coefficient about Body z-axis} \end{aligned}$$

where $\mathrm{C_N}$=-$\mathrm{C_Z}$ and $\mathrm{C_A}$=-$\mathrm{C_X}$. The more commonly used lift and drag coefficients (C_L, C_D) are geometrically related to the axial and the normal force coefficients (C_A, C_N) and can be determined by

$$F_A^w = \begin{bmatrix} -D \\ -S \\ -L \end{bmatrix} = \left[T\right]^{wb} \mathbf{F}_A^b = \left[T\right]^{wb} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.77}$$

where $\left[T\right]^{wb}$ is the transformation matrix that rotates the body-frame to the wind-frame through the aerodynamic angles and is given as

$$\left[T\right]^{wb} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.78}$$

This yields the following expressions for drag, sideforce, and lift coefficients

$$\mathrm{C_D} = -\mathrm{C_X}\cos\alpha\cos\beta - \mathrm{C_Y}\sin\beta - \mathrm{C_Z}\sin\alpha\cos\beta \tag{3.79}$$

$$C_S = C_X \cos\alpha \sin\beta - C_Y \cos\beta + C_Z \sin\alpha \sin\beta \tag{3.80}$$

$$C_L = C_X \sin\alpha - C_Z \cos\alpha \tag{3.81}$$

In terms of the sign convention along the body-axes (C_N=-C_Z and C_A=-C_X) these equations can be re-written as

$$C_D = C_A \cos\alpha \cos\beta - C_Y \sin\beta + C_N \sin\alpha \cos\beta \tag{3.82}$$

$$C_S = C_A \cos\alpha \sin\beta - C_Y \cos\beta - C_N \sin\alpha \sin\beta \tag{3.83}$$

$$C_L = -C_A \sin\alpha + C_N \cos\alpha \tag{3.84}$$

For zero sideslip $(\beta = 0)$, these equations reduce to the following

$$C_D = C_A \cos\alpha + C_N \sin\alpha \tag{3.85}$$

$$C_S = -C_Y \tag{3.86}$$

$$C_L = C_N \cos\alpha - C_A \sin\alpha \tag{3.87}$$

with their corresponding 2-D relation depicted in Figure 3.12.

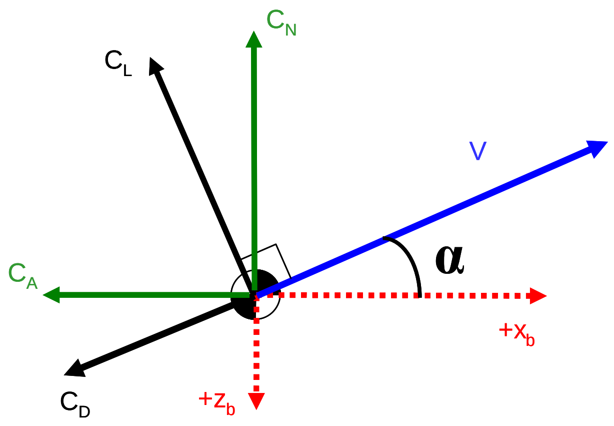


Figure 3.12 2-D Relationship Between Aerodynamic Coefficients for $\beta = 0$.

From Eq. (3.83), for angles of sideslip between 0-8 deg, $C_S \approx -C_Y$; therefore, for most flight regimes this is a valid assumption, especially for hypersonic reentry where sideslip is approximately zero for the majority of the flight.

(2) Stability and Control Derivatives. The stability and control derivatives, a result of small-disturbance (i.e., perturbation) equations, provide incremental changes to the aerodynamic coefficients as a result of changes in both longitudinal and lateral stability and control variables. That is, stability derivatives are

the partial derivatives of the force and moment coefficients with respect to various motion variables. For example, consider small-perturbations in force and represent the force equation for X as a Taylor series expansion [100],

$$\Delta X(u,\dot{u},w,\dot{w},...,\delta_e,\dot{\delta}_e) = \frac{\partial X}{\partial u}\Delta u + \frac{\partial X}{\partial \dot{u}}\Delta \dot{u} + ...\frac{\partial X}{\partial \delta_e}\Delta \delta_e + ...H.O.T. \quad (3.88)$$

Here, the stability derivative for the forward speed, u, is given by the partial derivative term $\frac{\partial X}{\partial u}$. This dimensional stability derivative can then be non-dimensionalized to yield its corresponding stability coefficient, C_{X_u}. The aero data used in this work *explicitly* depends on the non-dimensional longitudinal and lateral derivatives shown in Table 3.5 and Table 3.6, respectively.

Table 3.5 Nondimensional Longitudinal Derivatives

	C_A	C_N	C_m
α	C_{A_α}	C_{N_α}	C_{m_α}
u	C_{A_u}	C_{N_u}	C_{m_u}
$\hat{q}$	C_{A_q}	C_{N_q}	C_{m_q}

Table 3.6 Nondimensional Lateral Derivatives

	C_Y	C_l	C_n
β	C_{Y_β}	C_{l_β}	C_{n_β}
$\hat{p}$	C_{Y_p}	C_{l_p}	C_{n_p}
$\hat{r}$	C_{Y_r}	C_{l_r}	C_{n_r}

Note that the terms $\hat{p}$, $\hat{q}$, and $\hat{r}$ are the scaled body angular rates more thoroughly defined in the next section.

(3) Aerodynamic Data and Component Buildup. Aerodynamic coefficients are determined by various methods to include wind tunnel testing, flight testing, computational fluid dynamics (CFD) analysis, analytical analysis based on theoretical approximations, or a combination of such. A more detailed explanation of aerodynamic coefficient measurements and estimations can be found in Ref. [26]. When used for computer modeling and simulation applications, empirical data is often tabulated so that computer simulations can perform a series of calls to a collection of data tables. For the 6-DOF models in this work, aerodynamic data is tabulated such that it depends on 11 independent variables creating a significantly large database of tables that in itself, takes a noticeable amount of computer resources to retrieve a coefficient value. This is especially true for highly iterative algorithms such as those involving optimization routines.

In addition to the size of the tables, organization of the data is also a factor that can affect computational performance. Due to the discrete data, interpolation is required for arbitrary values throughout the vehicle's flight regime. Since it is difficult to physically interpret (i.e., visualize) higher-dimensional data, the task of developing an interpolating algorithm is further complicated. For this reason, it is advantageous to use a component buildup method such that the individual aero coefficients are formed from a sum of components. In this fashion, the measurement (via wind tunnel), organization of data, and computer/mathematical manipulation and interpolation is simplified. Individual components have fewer dimensions that are much easier to interpolate in a table or approximate by curve/surface fitting if needed.

To demonstrate the component buildup of a coefficient, consider only the axial force coefficient. First, the coefficient can be expressed as a sum of a base term and a delta term

$$C_A = C_{A_{BAE}} + C_{A_{\underline{\delta}}} \tag{3.89}$$

where $C_{A_{BAE}}$ is the base-aero-engine (BAE) contribution to axial force that includes wing-body and propulsion interference effects and $C_{A_{\underline{\delta}}}$ is the control effector vector contribution to the axial force that is a summation of the effects from the individual control surface deflections such that $C_{A_{\underline{\delta}}} = \sum_{i=1}^{n} C_{A_{\delta_i}}$ for the n-controls (n=8 for X-33). Secondly, the BAE contribution consists of a base term and a damping derivative term such that

$$C_{A_{BAE}} = C_{A_{base}} + C_{A_q}\hat{q} \tag{3.90}$$

where $\hat{q}$ is a scaled pitch rate. It is necessary to non-dimensionalize the damping derivatives using the normalized scaling factors from thin-airfoil theory: $(\frac{b}{2V})$ and $(\frac{\bar{c}}{2V})$ for lateral-directional and longitudinal terms, respectively, to form the scaled rates

$$\begin{aligned}\hat{p} &= \frac{pb}{2V}\\ \hat{q} &= \frac{q\bar{c}}{2V}\\ \hat{r} &= \frac{rb}{2V}\end{aligned} \tag{3.91}$$

such that the change in axial force coefficient due to pitch rate, C_{A_q}, is non-dimensionalized as follows

$$C_{A_q}\hat{q} = C_{A_q} q\left(\frac{\bar{c}}{2V}\right) \rightarrow \frac{1}{\text{rad}} \times \frac{\text{rad}}{\text{sec}} \times \left(\frac{\text{ft}}{\text{ft/sec}}\right) = \text{no units!} \tag{3.92}$$

Note that in terms of linear incremental changes, the base term can also be expressed as

$$C_{A_{base}} = C_{A_0} + C_{A_\alpha}\alpha + C_{A_\beta}\beta \tag{3.93}$$

where C_{A_0} is the axial force coefficient when the vehicle is in the "clean" configuration, that is, at zero AoA and zero bank angle with no control surfaces deflected; C_{A_α} is the axial force coefficient due to AoA; and C_{A_β} is the axial force coefficient due to bank angle. All together, each of the BAE terms for the aerodynamic coefficients are given as

$$
\begin{aligned}
\mathrm{C}_{\mathrm{A}_{\mathrm{BAE}}} &= \mathrm{C}_{\mathrm{A}_{\mathrm{base}}} + \mathrm{C}_{\mathrm{A}_{\hat{\mathrm{q}}}} \\
\mathrm{C}_{\mathrm{Y}_{\mathrm{BAE}}} &= \mathrm{C}_{\mathrm{Y}_{\mathrm{base}}} + \mathrm{C}_{\mathrm{Y}_{\hat{\mathrm{p}}}} + \mathrm{C}_{\mathrm{Y}_{\hat{\mathrm{r}}}} \\
\mathrm{C}_{\mathrm{N}_{\mathrm{BAE}}} &= \mathrm{C}_{\mathrm{N}_{\mathrm{base}}} + \mathrm{C}_{\mathrm{N}_{\hat{\mathrm{q}}}} \\
\mathrm{C}_{\mathrm{l}_{\mathrm{BAE}}} &= \mathrm{C}_{\mathrm{l}_{\mathrm{base}}} + \mathrm{C}_{\mathrm{l}_{\hat{\mathrm{r}}}} + \mathrm{C}_{\mathrm{l}_{\hat{\mathrm{p}}}} \\
\mathrm{C}_{\mathrm{m}_{\mathrm{BAE}}} &= \mathrm{C}_{\mathrm{m}_{\mathrm{base}}} + \mathrm{C}_{\mathrm{m}_{\hat{\mathrm{q}}}} \\
\mathrm{C}_{\mathrm{n}_{\mathrm{BAE}}} &= \mathrm{C}_{\mathrm{n}_{\mathrm{base}}} + \mathrm{C}_{\mathrm{n}_{\hat{\mathrm{r}}}} + \mathrm{C}_{\mathrm{n}_{\hat{\mathrm{p}}}}
\end{aligned}
\tag{3.94}
$$

such that the total aerodynamic coefficients can be represented as a component buildup with their respective dependencies listed in the following form

$$
\begin{aligned}
\mathrm{C}_{\mathrm{A}} &= \mathrm{C}_{\mathrm{A}_{\mathrm{base}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{A}_{\hat{\mathrm{q}}}}(\alpha,M,q) + \mathrm{C}_{\mathrm{A}_{\delta_{\mathrm{e}}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{A}_{\delta_{\mathrm{bf}}}}(\alpha,\beta,M) \\
\mathrm{C}_{\mathrm{Y}} &= \mathrm{C}_{\mathrm{Y}_{\mathrm{base}}}(\beta) + \mathrm{C}_{\mathrm{Y}_{\hat{\mathrm{p}}}}(\alpha,M,p) + \mathrm{C}_{\mathrm{Y}_{\hat{\mathrm{r}}}}(\alpha,M,r) + \mathrm{C}_{\mathrm{Y}_{\delta_{\mathrm{e}}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{Y}_{\delta_{\mathrm{r}}}}(\alpha,\beta,M) \\
\mathrm{C}_{\mathrm{N}} &= \mathrm{C}_{\mathrm{N}_{\mathrm{base}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{N}_{\hat{\mathrm{q}}}}(\alpha,M,q) + \mathrm{C}_{\mathrm{N}_{\delta_{\mathrm{e}}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{N}_{\delta_{\mathrm{bf}}}}(\alpha,\beta,M) \\
\mathrm{C}_{\mathrm{l}} &= \mathrm{C}_{\mathrm{l}_{\mathrm{base}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{l}_{\hat{\mathrm{r}}}}(\alpha,M,r) + \mathrm{C}_{\mathrm{l}_{\hat{\mathrm{p}}}}(\alpha,M,p) + \mathrm{C}_{\mathrm{l}_{\delta_{\mathrm{e}}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{l}_{\delta_{\mathrm{r}}}}(\alpha,\beta,M) \\
\mathrm{C}_{\mathrm{m}} &= \mathrm{C}_{\mathrm{m}_{\mathrm{base}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{m}_{\hat{\mathrm{q}}}}(\alpha,M,q) + \mathrm{C}_{\mathrm{m}_{\delta_{\mathrm{e}}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{m}_{\delta_{\mathrm{bf}}}}(\alpha,\beta,M) \\
\mathrm{C}_{\mathrm{n}} &= \mathrm{C}_{\mathrm{n}_{\mathrm{base}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{n}_{\hat{\mathrm{r}}}}(\alpha,M,r) + \mathrm{C}_{\mathrm{n}_{\hat{\mathrm{p}}}}(\alpha,M,p) + \mathrm{C}_{\mathrm{n}_{\delta_{\mathrm{e}}}}(\alpha,\beta,M) + \mathrm{C}_{\mathrm{n}_{\delta_{\mathrm{r}}}}(\alpha,\beta,M)
\end{aligned}
\tag{3.95}
$$

Now, substituting (3.90) and $C_{A_{\underline{\delta}}} = \sum_{i=1}^{n} C_{A_{\delta_i}}$, the aerodynamic force $X = X_{BAE} + X_{\underline{\delta}}$ in Eq. (3.54) becomes

$$X_{BAE} = -\overline{q}S_{ref}C_{A_{BAE}} = -\overline{q}S_{ref}\left\{C_{A_{base}} + C_{A_q}q\left(\frac{\overline{c}}{2V}\right)\right\} \tag{3.96}$$

$$X_{\underline{\delta}} = -\overline{q}S_{ref}C_{A_{\underline{\delta}}} = -\overline{q}S_{ref}\left\{\sum_{i=1}^{8} C_{A_{\delta_i}}\right\} \tag{3.97}$$

where $\overline{q}$ is the dynamic pressure given as

$$\overline{q} = \frac{1}{2}\rho(z)V^2 \tag{3.98}$$

(4) Aerodynamic Approximations via Data Fitting. Table look-up data is inherently non-smooth with numerous discontinuities, especially so for uncertain aerodynamic data. Measured data from flight-tests and/or wind-tunnel tests is always subject to error sources such as instrumentation inaccuracies, human error, etc. For example, the following coefficient uncertainties for the HL-20 lifting-body

configuration are primary influenced by calibration accuracy of a strain gage used during wind-tunnel testing [172].

C_N	C_A	C_Y	C_m	C_n	C_l
±0.00352	±0.000264	±0.00044	±0.00090	±0.00029	±0.000156

Note that specific comments about X-33 aero-model uncertainty are presented in the 6-DOF results (Chapter VI) and Appendix A.

From a statistical perspective, large errors may result from an inadequate number of test runs, like not performing enough flight test maneuvers or running enough wind tunnel tests. Likewise, engineering code estimates often have errors due to variations in maneuver quality, analysis, engineering judgment, and prediction techniques [173]. A NASA study, reported by Cobleigh, developed an X-33 aerodynamic uncertainty model that indicated up to 43% uncertainty in the pitching moment due to body flap deflections and 80% uncertainty due to pitch damping [173]. In terms of data organization, another source of error is caused by piecing together data from different test runs and from sensitivities to small changes in the independent variables [26]. Although using a component buildup scheme seems to simplify data handling, any problems with the data are propagated by each component. For these reasons and the reasons mentioned in the previous section, aerodynamic data modeling is an important concern for flight simulations, especially when solution accuracy is paramount. The two primary approaches for improving aero-data models are data fitting via curve-fitting approximations or data smoothing via regression analysis.

Various methods can be used to smooth and re-tabulate the data such as polynomial approximations, splines, etc. Interpolation, smoothing methods, neural networks, and numerical results for handling large data tables and their effects on computational performance are mentioned as possibilities for future work in Chapter VI.

e. Complete Standard 6-DOF EoM

Now that the position, velocity, body-rate, and Euler angle equations have been presented and defined, the complete 6-DOF EoM can be succinctly combined as shown in Figure 3.13.

$$\dot{x}_e = [\cos\theta\cos\psi]V\cos\alpha\cos\beta + [\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi]V\sin\beta + [\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi]V\sin\alpha\cos\beta$$

$$\dot{y}_e = [\cos\theta\sin\psi]V\cos\alpha\cos\beta + [\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi]V\sin\beta + [\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi]V\sin\alpha\cos\beta$$

$$\dot{z}_e = [-\sin\theta]V\cos\alpha\cos\beta + [\sin\phi\cos\theta]V\sin\beta + [\cos\phi\cos\theta]V\sin\alpha\cos\beta$$

$$\dot{V} = \left[\frac{X}{m} - g\sin\theta\right]\cos\alpha\cos\beta + \left[\frac{Y}{m} + g\sin\phi\cos\theta\right]\sin\beta + \left[\frac{Z}{m} + g\cos\phi\cos\theta\right]\sin\alpha\cos\beta$$

$$\dot{\alpha} = \frac{1}{mV\cos\beta}\left[Z\cos\alpha - X\sin\alpha\right] + \frac{g}{V\cos\beta}\left[\cos\phi\cos\theta\cos\alpha + \sin\theta\sin\alpha\right] + q - \tan\beta\left[p\cos\alpha + r\sin\alpha\right]$$

$$\dot{\beta} = \frac{\cos\beta}{mV}\left[Y + mg\sin\phi\cos\theta\right] + p\sin\alpha - r\cos\alpha - \frac{\sin\beta}{mV}\left[Z\sin\alpha + X\cos\alpha\right] - \frac{g\sin\beta}{V}\left[\cos\phi\cos\theta\sin\alpha - \sin\theta\cos\alpha\right]$$

$$\dot{p} = \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}^{2}}\left[L - \left(I_{zz} - I_{yy}\right)qr + I_{xz}qp\right] + \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^{2}}\left[N + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right]$$

$$\dot{q} = \frac{1}{I_{yy}}\left[M - \left(I_{xx} - I_{zz}\right)pr - I_{xz}\left(p^2 - r^2\right)\right]$$

$$\dot{r} = \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^{2}}\left[L + \left(I_{yy} - I_{zz}\right)qr + I_{xz}qp\right] + \frac{I_{x}}{I_{xx}I_{zz} - I_{xz}^{2}}\left[N + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right]$$

$$\dot{\phi} = p + \left(q\sin\phi + r\cos\phi\right)\tan\theta$$

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

$$\dot{\psi} = \left(q\sin\phi + r\cos\phi\right)\sec\theta$$

$$\boxed{\begin{aligned}
X &= \frac{1}{2}\rho(z)V^2 S_{ref} C_X\left(\alpha,\beta,M,\underline{\delta}\right)\\
Y &= \frac{1}{2}\rho(z)V^2 S_{ref} C_Y\left(\beta,M,\underline{\delta}\right)\\
Z &= \frac{1}{2}\rho(z)V^2 S_{ref} C_Z\left(\alpha,\beta,M,\underline{\delta}\right)\\
L &= \frac{1}{2}\rho(z)V^2 S_{ref} b C_l\left(\beta,M,\underline{\delta}\right)\\
M &= \frac{1}{2}\rho(z)V^2 S_{ref} \overline{c} C_m\left(\alpha,\beta,M,\underline{\delta}\right)\\
N &= \frac{1}{2}\rho(z)V^2 S_{ref} b C_n\left(\beta,M,\underline{\delta}\right)
\end{aligned}}$$

Figure 3.13 Standard 6-DOF EoM for Flat Earth, No Wind Model.

These standard 6-DOF equations assume a rigid-vehicle, constant mass, constant inertia tensor, no thrust, no wind, non-relativistic mechanics, flat-earth model, and negligible cross-products of inertia for I_{xy} and I_{yz}. Although common in aircraft simulations, these equations may pose various problems for G&C applications involving trajectory optimization schemes, especially for hypersonic reentry guidance.

f. Drawbacks of the Standard Flight EoM

Despite their relatively straight-forward derivation and common use in modeling and simulation studies, the standard EoM have various drawbacks that may prohibit their use in trajectory optimization applications. Among these drawbacks are complexity, singularities, and scaling.

By visually inspecting the complete equations in Figure 3.13, it is obvious that they are quite complex. The position equations $(\dot{x}_e, \dot{y}_e, \dot{z}_e)$ alone consist of numerous nonlinear trigonometric functions that are highly coupled to both the velocity equations $(\dot{V}, \dot{\alpha}, \dot{\beta})$ and the rotational Euler-rate equations $(\dot{\phi}, \dot{\theta}, \dot{\psi})$. From a modeling approach, this highly-coupled nature of the translational and rotational equations poses a problem with using a homotopy path to problem solving (as discussed later). These equations also present potential singularity problems. In addition to the singularity condition in the Euler-rate equations when $\theta = \pm 90\,\text{deg}$, the spherical velocity equations also have singularities at $V = 0$ and $\beta = \pm 90\,\text{deg}$. As mentioned earlier, an alternate formulation that avoids the singularities associated with the spherical velocity equations consists of using the linear velocity equations, see (3.53), repeated here for clarity,

$$\begin{aligned}
\dot{u} &= \frac{X}{m} + g\sin\theta + rv - qw \\
\dot{v} &= \frac{Y}{m} - g\cos\theta\sin\phi - ru + pw \\
\dot{w} &= \frac{Z}{m} - g\cos\theta\cos\phi + qu - pv
\end{aligned} \tag{3.99}$$

and replacing the corresponding terms in the position equations with (3.26),

$$\begin{aligned}
u &= V\cos\alpha\cos\beta \\
v &= V\sin\beta \\
w &= V\sin\alpha\cos\beta
\end{aligned} \tag{3.100}$$

to revert back to (3.29),

$$\begin{aligned}
\dot{x}_E &= u(\cos\theta\cos\psi) + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) \\
&\quad + w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\
\\
\dot{y}_E &= u(\cos\theta\sin\psi) + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) \\
&\quad + w(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\
\\
\dot{z}_E &= u(-\sin\theta) + v(\sin\phi\cos\theta) + w(\cos\phi\cos\theta)
\end{aligned} \tag{3.101}$$

The drawback of using these alternate equations for position and velocity are that AoA and sideslip angle are not readily available (i.e., not part of the state vector). Since they are required for aerodynamic force and moment calculations, they would have to be computed using Eqs. (3.33) and (3.34). Also, substituting (3.99) and (3.101) into the standard equations provides slight simplifications without as many singularities but, there are still drawbacks from the computational perspective. Recall from Chapter II that scaling and balancing equations can have a significant effect on the computational efficiency of a numerical algorithm. As previously pointed out, using x,y,z-Cartesian positional coordinates in the body-fixed frame can present a scaling problem for flight trajectories involving drastically different crossrange and downrange values. This scaling issue can be avoided altogether by selecting coordinates that are more advantageous for numerical methods such as the spherical polar coordinates in the vehicle-carried frame.

One final drawback to the standard equations is their limited use for applications involving high-speed flight over large portions of the earth's surface such as the case for reentry problems. As mentioned, the Cartesian position equations assume a locally flat earth such that the earth-fixed inertial frame is equivalent to the local horizontal frame. To account for the spherical, rotating earth effects encountered in reentry problems, additional terms must be added to the translational EoM. Using the standard equations requires these new spherical terms be converted to a Cartesian representation that adds additional complexity. Therefore, it makes sense that using spherical polar coordinates in the first place will eliminate any unnecessary complexities and potential computational problems.

2. Alternate Flight Vehicle Model(s)

A much less complex and numerically tractable formulation exists using spherical polar coordinates in the vehicle-carried frame. Substituting (3.23) for (3.31), and taking the derivatives of Eqs. (3.41)-(3.43) with respect to time yields the following position and velocity equations

$$\begin{aligned}
\dot{R} &= V\sin\gamma \\
\dot{\mu} &= \frac{V}{R\cos\lambda}\cos\gamma\sin\xi \\
\dot{\lambda} &= \frac{V}{R}\cos\gamma\cos\xi \\
\dot{V} &= \frac{1}{m}(\mathbf{Y}\sin\beta - \mathbf{D}\cos\beta) - g\sin\gamma \\
\dot{\gamma} &= \frac{1}{mV}(\mathbf{L}\cos\sigma - \mathbf{D}\sin\beta\sin\sigma - \mathbf{Y}\cos\beta\sin\sigma) \\
&\quad + (\frac{V}{r} - \frac{g}{V})\cos\gamma \\
\dot{\xi} &= \frac{1}{mV\cos\gamma}(\mathbf{L}\sin\sigma + \mathbf{D}\sin\beta\cos\sigma + \mathbf{Y}\cos\beta\cos\sigma) \\
&\quad - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda
\end{aligned} \tag{3.102}$$

where according to Eqs. (3.79)-(3.81), the X, Y, and Z forces have been replaced by the lift, drag, and side forces represented in Eq. (3.102) as **L**, **D**, **S**, respectively. With the addition of spherical, rotating earth terms, these equations alone constitute the standard 3-DOF point-mass EoM presented by Vinh, Buseman, and Culp [95], as shown in Figure 3.16(a). Now, the original 6-DOF rotational equations can be appended to (3.102) to give a mixed body-frame and air-relative frame set of EoM succinctly written as

$$\dot{R} = V \sin\gamma$$

$$\dot{\mu} = \frac{V}{R\cos\lambda}\cos\gamma\sin\xi$$

$$\dot{\lambda} = \frac{V}{R}\cos\gamma\cos\xi$$

$$\dot{V} = \frac{1}{m}(\mathbf{Y}\sin\beta - \mathbf{D}\cos\beta) - g\sin\gamma$$

$$\dot{\gamma} = \frac{1}{mV}(\mathbf{L}\cos\sigma - \mathbf{D}\sin\beta\sin\sigma - \mathbf{Y}\cos\beta\sin\sigma) + (\frac{V}{r} - \frac{g}{V})\cos\gamma$$

$$\dot{\xi} = \frac{1}{mV\cos\gamma}(\mathbf{L}\sin\sigma + \mathbf{D}\sin\beta\cos\sigma + \mathbf{Y}\cos\beta\cos\sigma) - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda$$

$$\dot{p} = \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}^2}\left[L - \left(I_{zz} - I_{yy}\right)qr + I_{xz}qp\right] + \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2}\left[N + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right]$$

$$\dot{q} = \frac{1}{I_{yy}}\left[M - \left(I_{xx} - I_{zz}\right)pr - I_{xz}\left(p^2 - r^2\right)\right]$$

$$\dot{r} = \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2}\left[L + \left(I_{yy} - I_{zz}\right)qr + I_{xz}qp\right] + \frac{I_{x}}{I_{xx}I_{zz} - I_{xz}^2}\left[N + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right]$$

$$\dot{\phi} = p + \left(q\sin\phi + r\cos\phi\right)\tan\theta$$

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

$$\dot{\psi} = \left(q\sin\phi + r\cos\phi\right)\sec\theta$$

Figure 3.14 Alternate 6-DOF EoM.

3. Contrast Standard vs. Alternate Model

Comparing the standard EoM in Figure 3.13 with the alternate set in Figure 3.14, it is apparent that the complexity associated with the first six equations is removed. This simplification is a direct result of eliminating the Euler angles as the primary coupling terms. Instead, the primary coupling of the translational and rotational equations is through the flight angles α and β. Note that only β is explicit in the equations; whereas both α and β are embedded in the aerodynamic forces and moments by the inter-dependence of the aerodynamic coefficients, e.g.,, $C_Z = f(\alpha, \beta, M, \underline{\delta})$. Like the alternate position and velocity equations given for the standard model, Eqs. (3.99) and (3.100), this model also requires α and β to be calculated indirectly; however, instead of the remedial calculations of Eqs. (3.33) and (3.34), the more accurate relations given by Eqs. (3.16)-

(3.18) can be used. This EoM selection naturally removes the scaling problem presented by the crossrange position equation, $\dot{y}_E$, in the standard EoM.

E. REDUCED-ORDER DYNAMICAL MODEL (3-DOF)

Most reduced-order models ($\leq 4\text{-DOF}$) assume the vehicle is a point-mass as opposed to a rigid body. This means that only the translational equations are implemented without regards to the rotational equations as included in the 6-DOF formulations in Sec. D.

1. Point-Mass Equations with Sideslip

Commonly used sets of dynamical equations of motion for an un-powered reentry vehicle are shown in Figure 3.15, illustrating the simplification from a spherical, rotating-earth model to a flat-earth model. Note that in these equations, the Heading/Azimuth is measured positive clock-wise from the East as opposed to the conventional positive clockwise from the North.

$$
\begin{aligned}
\dot{r} &= V\sin\gamma \\
\dot{\mu} &= \frac{V\cos\gamma\cos\xi}{r\cos\lambda} \\
\dot{\lambda} &= \frac{V\cos\gamma\sin\xi}{r} \\
\dot{V} &= \frac{1}{m}(Y\sin\beta - D\cos\beta) - g\sin\gamma \\
&\quad + \Omega^2 r\cos\lambda(\sin\gamma\cos\lambda - \cos\gamma\sin\lambda\sin\xi) \\
\dot{\gamma} &= \frac{1}{mV}(L\cos\sigma - D\sin\beta\sin\sigma - Y\cos\beta\sin\sigma) \\
&\quad + (\frac{V}{r} - \frac{g}{V})\cos\gamma + 2\Omega\cos\lambda\cos\xi \\
&\quad + \frac{\Omega^2 r}{V}\cos\lambda(\cos\gamma\cos\lambda + \sin\gamma\sin\lambda\sin\xi) \\
\dot{\xi} &= \frac{1}{mV\cos\gamma}(L\sin\sigma + D\sin\beta\cos\sigma + Y\cos\beta\cos\sigma) \\
&\quad - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda + 2\Omega(\tan\gamma\cos\lambda\sin\xi - \sin\lambda) \\
&\quad - \frac{\Omega^2 r}{V\cos\gamma}\sin\lambda\cos\lambda\cos\xi
\end{aligned}
$$

(a) Spherical, Rotating Earth

$(\Omega \neq 0)$

$\Rightarrow$

$$
\begin{aligned}
\dot{r} &= V\sin\gamma \\
\dot{\mu} &= \frac{V\cos\gamma\cos\xi}{r\cos\lambda} \\
\dot{\lambda} &= \frac{V\cos\gamma\sin\xi}{r} \\
\dot{V} &= \frac{1}{m}(Y\sin\beta - D\cos\beta) - g\sin\gamma \\
\dot{\gamma} &= \frac{1}{mV}(L\cos\sigma - D\sin\beta\sin\sigma - Y\cos\beta\sin\sigma) \\
&\quad + (\frac{V}{r} - \frac{g}{V})\cos\gamma \\
\dot{\xi} &= \frac{1}{mV\cos\gamma}(L\sin\sigma + D\sin\beta\cos\sigma + Y\cos\beta\cos\sigma) \\
&\quad - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda
\end{aligned}
$$

(b) Spherical, Non-Rotating Earth

$(\Omega = 0)$

$\Downarrow$

$$
\begin{aligned}
\dot{x}_E &= V\cos\gamma\cos\xi \\
\dot{y}_E &= V\cos\gamma\sin\xi \\
\dot{z}_E &= V\sin\gamma \\
\dot{V} &= \frac{1}{m}(Y\sin\beta - D\cos\beta) - g\sin\gamma \\
\dot{\gamma} &= \frac{1}{mV}(L\cos\sigma - D\sin\beta\sin\sigma - Y\cos\beta\sin\sigma) - \frac{g}{V}\cos\gamma \\
\dot{\xi} &= \frac{1}{mV\cos\gamma}(L\sin\sigma + D\sin\beta\cos\sigma + Y\cos\beta\cos\sigma)
\end{aligned}
$$

(c) Flat Earth

Figure 3.15 Common Point-Mass Equations of Motion ($\beta \neq 0$).

2. Point-Mass Equations with Zero Sideslip

Most applications assume steady, coordinated turns such that the sideslip angle is zero ($\beta = 0$). This further simplifies the equations as depicted in Figure 3.16.

$$
\begin{aligned}
\dot{r} &= V\sin\gamma \\
\dot{\mu} &= \frac{V\cos\gamma\cos\xi}{r\cos\lambda} \\
\dot{\lambda} &= \frac{V\cos\gamma\sin\xi}{r} \\
\dot{V} &= \frac{-D}{m} - g\sin\gamma \\
&\quad + \Omega^2 r\cos\lambda(\sin\gamma\cos\lambda - \cos\gamma\sin\lambda\sin\xi) \\
\dot{\gamma} &= \frac{L\cos\sigma}{mV} + (\frac{V}{r} - \frac{g}{V})\cos\gamma + 2\Omega\cos\lambda\cos\xi \\
&\quad + \frac{\Omega^2 r}{V}\cos\lambda(\cos\gamma\cos\lambda + \sin\gamma\sin\lambda\sin\xi) \\
\dot{\xi} &= \frac{L\sin\sigma}{mV\cos\gamma} - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda \\
&\quad + 2\Omega(\tan\gamma\cos\lambda\sin\xi - \sin\lambda) \\
&\quad - \frac{\Omega^2 r}{V\cos\gamma}\sin\lambda\cos\lambda\cos\xi
\end{aligned}
\quad \Rightarrow
$$

(a) Spherical, Rotating Earth

$(\Omega \neq 0)$

$$
\begin{aligned}
\dot{r} &= V\sin\gamma \\
\dot{\mu} &= \frac{V\cos\gamma\cos\xi}{r\cos\lambda} \\
\dot{\lambda} &= \frac{V\cos\gamma\sin\xi}{r} \\
\dot{V} &= \frac{-D}{m} - g\sin\gamma \\
\dot{\gamma} &= \frac{L\cos\sigma}{mV} + (\frac{V}{r} - \frac{g}{V})\cos\gamma \\
\dot{\xi} &= \frac{L\sin\sigma}{mV\cos\gamma} - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda
\end{aligned}
\quad \Rightarrow
$$

(b) Spherical, Non-Rotating Earth

$(\Omega = 0)$

$$
\begin{aligned}
\dot{x}_E &= V\cos\gamma\cos\xi \\
\dot{y}_E &= V\cos\gamma\sin\xi \\
\dot{z}_E &= V\sin\gamma \\
\dot{V} &= \frac{-D}{m} - g\sin\gamma \\
\dot{\gamma} &= \frac{L\cos\sigma}{mV} - \frac{g}{V}\cos\gamma \\
\dot{\xi} &= \frac{L\sin\sigma}{mV\cos\gamma}
\end{aligned}
$$

(c) Flat Earth

Figure 3.16 Common Point-Mass Equations of Motion ($\beta = 0$)

3. General Comments Regarding Point-Mass Models

The subtle effects associated with a spherical, rotating Earth are represented by the Coriolis acceleration term (2Ω), the centrifugal acceleration term ($\Omega^2 r$), and the variation of gravitational acceleration ($r = R_E + h$) in Figure 3.15(a) and Figure 3.16(a). For high-speed, long-range flights over the surface of the Earth, these *round-earth effects* should not be ignored for high-accuracy analysis. On the other hand, when concerned with only local effects or variation of the vehicle's suborbital speed and altitude over a certain (shorter) range of the flight trajectory, a flat-earth model is viable in which the corresponding terms vanish as in Figure 3.15(c) and Figure 3.16(c). When transforming from (b) to (c) in both figures, recall that the conversion of spherical kinematics to Cartesian ("local-horizontal") kinematics was demonstrated in Sec.D.1.a.

F. MODELING THE ENVIRONMENT

As previously indicated, it is paramount that accurately solving a problem requires the correct formulation to the degree of fidelity dictated by the specific application. Likewise, an important aspect of formulating any engineering problem is the underlying assumptions made. In most cases, assumptions are made in an effort to simplify the problem. For solving problems that seek high-accuracy results, careful choices must be made when deciding, if at all, on simplifying assumptions. For the modeling and simulation of aerospace vehicles, the fidelity of the system model depends heavily on the fidelity of the subsystem models. Of primary importance is the choice of subsystem models that approximate the environment in which the vehicle will be operating (i.e., the flight envelope). For the reentry problem, high-speed, trans-atmospheric flight over large areas of the Earth's surface necessitates the use of an Earth model, including its shape, rotation, and gravity, and an atmospheric model. The following sections give a brief description of the environment models of varying fidelity.

1. Earth Model: Shape and Gravity

As with any model, design or selection depends on the specific application and the desired accuracy of the results. For the Earth model, of particular interest is the trade-offs between flat vs. round shape, including spherical vs. oblate spheroid, non-rotating vs. rotating, and gravity based on potential function, inverse-gravity relation, or constant vector. The trade-offs are discussed in subsequent sections, but for now, a description of the models are provided for clarity later. Note that extensive details on Earth's form and its gravitational field can be found in Ref. [96].

a. Ellipsoidal-Earth Model

The highest fidelity Earth model is based on the ellipsoidal approximation provided by the 1984 World Geodetic System (WGS-84) [174]. The irregular shape of the Earth, *geoid*, is modeled as an ellipsoid of revolution (oblate spheroid) defined by the primary characteristics of semi-major, a, and semi-minor axes, b; flattening factor or ellipticity, f; eccentricity, e; sidereal rate of rotation, ω_E; and gravitational constant, GM, all given as:

$$a = 6{,}378{,}137 \text{ m}$$

$$b = 6{,}356{,}752 \text{ m}$$

$$f = \frac{a-b}{a} \approx 3.35281066475 \times 10^{-3}$$

$$e = \frac{(a^2 - b^2)^{1/2}}{a} \approx 0.08181919$$

$$\omega_E = 7.2921150 \times 10^{-5} \text{ rad/s}$$

$$GM \equiv \mu = 3.986004418 \times 10^{14} \text{ m}^3/\text{s}^2$$

Although there is a difference between *geometrical* and *physical geodesy*, all 6-DOF models suffice with the WGS-84 approximation [97]. Note that the semi-major axis, *a*, is often referenced as Earth's equatorial radius, $R_E = 6378.136\,\text{km}$. Also, ω_E assumes a constant rate of rotation around a fixed inertial axis.

The gravity model that corresponds with the precision of WGS-84 is the 1996 Earth Gravitational Model (EGM96) that is based on gradients of a scalar potential function [175]. The gradients of a simplified potential function [26] are given in geocentric coordinates as

$$\mathbf{G}_E = \frac{GM}{r^2}\begin{bmatrix} 3J_2(a/r)^2 \sin\lambda_c \cos\lambda_c \\ 1 + 1.5J_2(a/r)^2(3\sin^2\lambda_c - 1) \end{bmatrix} \tag{3.103}$$

with $J_2 = \sqrt{5}\bar{C}_{2,0} = -1.0826267 \times 10^{-3}$ as the zonal harmonic term that includes the largest potential function coefficient, $\bar{C}_{2,0}$. According to Stevens and Lewis, there are hundreds-of-thousands of coefficients, but using only $\bar{C}_{2,0}$ "removes the dependence on terrestrial longitude" and results in extremely accurate models [26]. Also, it is possible that accurate accelerometers can compensate for the small errors associated with omitting higher harmonic terms.

b. Spherical-Earth Model

A medium-fidelity Earth model is then naturally defined as a spherical approximation to the oblate spheroid with the mean-radius defined as $R_m = 6367.474\ \text{km}$. Note that this radius is lower than the equatorial radius of the elliptical model.

The gravity field can be approximated from Newton's law of gravitational acceleration defined by the attracting force, *F*, between two objects with mass *M* and *m*, and separated by the radial distance *r*, such that

$$F = G\frac{Mm}{r^2} \tag{3.104}$$

where *G* is the universal gravitational constant. Solving this equation for *g* results in the well known *inverse-gravity law* stated as

$$g = \frac{GM}{r^2} \equiv \frac{\mu}{r^2} \tag{3.105}$$

The distance *r* is measured from the Earth's center of mass (cm) to the object's cm such that

$$r = R_m + h(z) \tag{3.106}$$

where $h(z)$ is the *geopotential altitude* defined as $h(z) = \dfrac{R_m z}{R_m + z}$ that accounts for the Earth's mass attraction and the centrifugal effect of its rotation. For flights remaining at lower altitudes, the geometric altitude, *z,* is adequate and can replace $h(z)$ in Eq. (3.106). Note that by setting $J_2 = 0$ in the ellipsoidal-earth gravity model, Eq. (3.103) reduces to the inverse-gravity relation essentially the same as setting $f = 0$ and $a = R_m$.

c. Flat-Earth Model

The lowest fidelity Earth model is the flat-earth approximation whereby Earth is considered a stationary plane of symmetry typically described by Cartesian x,y,z-coordinates. As such, this model neglects variations in gravity and rotational effects (i.e., Coriolis and centripetal/centrifugal). For this model, it is common to use the down component of the gravity vector at Earth's surface (45° geodetic latitude) given as $g = 9.806\ \text{m/s}^2$ and the distance *r* becomes the geometric distance *z*, now only the vertical distance above the flat-earth. In addition to the shape of the earth and the gravity field, another important part of the environment for a flight vehicle is the atmosphere.

2. Atmospheric Model

The Earth's atmosphere is defined by the fundamental fluid properties of pressure, density, and temperature that vary with altitude, time-of-day, weather, etc. and are related by the "equation-of-state." Since aerodynamic forces directly depend on air density, it is important for aeronautical engineering problems to use a standard model of the atmosphere.

a. Standard Atmosphere Tables

Various models exist based on measured, averaged, and curve-fitted data. Since the atmosphere is constantly changing, these models are updated every few years or so. Often used is the 1976 *U.S. Standard Atmosphere* that is normalized to "standard" sea-level conditions. Although it covers altitudes up to 1000 km at 45 degrees latitude, an accurate model above 80 km is irrelevant since the aerodynamic loads are negligible in this region [97]. Another common, but slightly out-dated model is the 1959 ARDC. Zipfel compares different models and discusses their subtle differences [97].

b. Atmosphere Approximations: Curve-Fits

Although many applications use variants of atmosphere tables developed over the years including curve-fits to the tabulated data as opposed to implementing strict table "look-ups", there are relatively accurate approximations that do not require actual measured atmosphere data. For example, a common curve-fit approximation to the atmosphere is the two-parameter exponential model that has many variants depending on the atmospheric region of interest. For closed-form solutions with altitudes between 5 km to 40 km, a good exponential approximation is given by Ref. [96] as

$$\rho(z) = \rho_0 e^{-h(z)/z_0} \tag{3.107}$$

where ρ_0 is the reference density taken to be 1.725 kg/m^3 (0.003399 slugs/ft^3), z is the current altitude, and $z_0 = \dfrac{RT}{g_0}$ is the atmospheric scale height (i.e., reference height) taken to be 6700 m (21981.6 ft). Note that R_m in the expression for $h(z)$ can be replaced by R_E for an elliptical-earth model, but there is negligible difference. Assuming the atmosphere is isothermal and sea-level temperature, the following values can be substituted: $\rho_0 = 1.225\ \text{kg/m}^3$ and $z_0 = 8434\,\text{m}$. An alternative to Eq. (3.107) is the variant

$$\rho(r) = \rho_0 e^{-\beta(r-r_0)} \tag{3.108}$$

where ρ_0 is the reference sea-level density, r is the current altitude measured from the center of the earth, β = 4.20168e-5 ft^{-1} is the inverse atmospheric scale height, and r_0 = 2.09e7 ft is a reference altitude. Note that when $r_0 = R_E \approx 20925646.3255$ ft then Eq. (3.108) is equivalent to Eq. (3.107). Therefore, with knowledge of the vehicle's operating envelope, this exponential model allows one to specify the altitude range of interest by selecting r_0 accordingly. Figure 3.17 compares two tabulated and two exponential atmosphere models. As seen, all converge at higher altitudes with slight discrepancies at the lower altitudes.

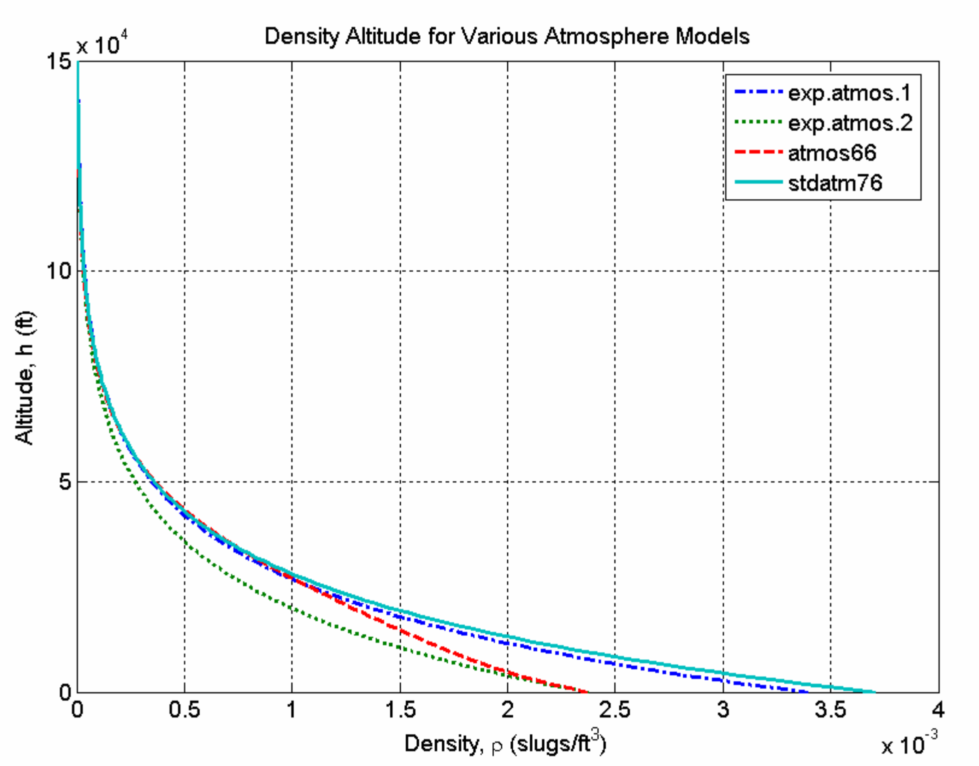


Figure 3.17 Comparison of Density-Altitude for Various Atmosphere Models.

3. Wind Model

The most common form of external disturbance encountered during atmospheric flight is the changing direction and intensity of the air mass itself (i.e., wind). There are various forms of wind and numerous ways to model wind.

a. Wind Gradient Approximation in Equations of Motion

Typical aircraft and wind disturbance analysis employs an analytical approximation to the wind's flow field characteristics. For example, a windshear/downdraft can be modeled as

$$W_x = wA(x) \tag{3.109}$$

$$W_h = w\frac{h}{h_*}B(x) \tag{3.110}$$

where *w* is the intensity of the windshear/downdraft combination; the functions *A(x)* and *B(x)* represent the wind gradients as shown in Figure 3.18; and h^* is a reference altitude based on the range of validity. At lower altitudes, these wind velocity components typically form a wind shear that is commonly represented by a parabolic gradient as a function of altitude.

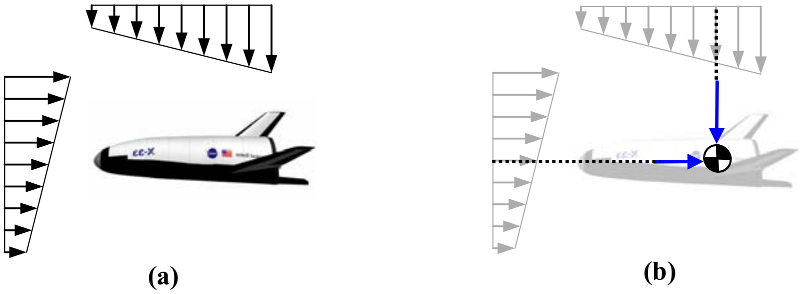


Figure 3.18 Approximation of Horizontal and Vertical Wind Gradients Acting on Vehicle.

Various wind models (both discrete and continuous) and modeling standards exist, such as numerous wind gust models, a standard Dryden Wind Turbulence Model, and a standard Von Karman Wind Turbulence Model as specified by Military Specification MIL-F-8785C [176] and Military Handbook MIL-HDBK-1797 [177]. As a feasibility study only, the work presented in Chapter IV employs a simplified approach to model a wind gust as illustrated by Figure 3.18 (b).

Assuming a flat-earth approximation to the equations of motion, wind is often modeled by adding wind velocity components in the Cartesian x,y,z-directions $\left(W_x, W_y, W_z\right)$ to the kinematical equations such that

$$\begin{aligned}\dot{x} &= V\cos\gamma\cos\xi + W_x \\ \dot{y} &= V\cos\gamma\sin\xi + W_y \\ \dot{z} &= V\sin\gamma + W_z\end{aligned} \tag{3.111}$$

and likewise, the wind acceleration components $\left(\dot{W}_x, \dot{W}_y, \dot{W}_z\right)$ are added to the dynamical equations of motion

$$\begin{aligned}\dot{V} &= -\frac{D(\alpha,V,z)}{m} - g(z)\sin\gamma - \left(\dot{W}_x\cos\gamma + \dot{W}_z\sin\gamma\right) \\ \dot{\gamma} &= \frac{L(\alpha,V,z)\cos\sigma}{mV} - \frac{g(z)\cos\gamma}{V} + \frac{1}{V}\left(\dot{W}_x\sin\gamma - \dot{W}_z\cos\gamma\right) \\ \dot{\xi} &= \frac{L(\alpha,V,z)\sin\sigma}{mV\cos\gamma}\end{aligned} \tag{3.112}$$

that when combined, form a new set of equations that includes the effects of a wind gradient model.

G. HOMOTOPY PATH TO MODELING

1. Model Reduction Process

Model fidelity has to do with the degree to which the model reproduces the essential characteristics of the real system being modeled such that a high-fidelity model is nearly indistinguishable from the real system. In the past, reducing the fidelity of a model by way of various simplifying assumptions was required in order to solve complex problems. One such process involved reducing the order of the problem in terms of the number of differential equations. Although reasons for using lower fidelity models for aerospace problem solving have shifted from mandatory to preliminary, the process is still required. Today, solving lower-fidelity problems is often used for preliminary analysis and for approximating solutions to higher order problems. These approximations can then be used as baselines for validating higher order solutions. The difficulty lies in the fact that experience and sound engineering judgment must go into the simplifying process. Also, there are some problems where low-order solutions are not representative of high-order solutions and care must be given as to how the low-order solution is interpreted and more importantly, how it is used. For example, a motivation for the high-fidelity reentry research presented in this work is that reentry algorithms based on low-fidelity models pose a potential safety concern. In this case, the two primary questions become: (1) what level of fidelity is adequate in representing the real system or in satisfying the intent of the problem? and (2) what solution gives trust worthy, meaningful results? For example, some engineers solving path planning problems base their models only on kinematical equations of motion. This may be suitable for generating a feasible path in terms of vehicle position and velocity, but by excluding the dynamics, the path may not be physically realizable in terms of forces and moments. The key is to base the simplification on a successive process of reducing the mathematical order of the system while still maintaining physical coherency. If nothing else, this homotopy path to modeling and problem solving is essential for understanding the system and debugging any mathematical, physical, or algorithmical problems.

2. General Homotopy Path to Modeling

The general homotopy path to modeling consists of defining the highest-order possible for a given system, say order *n*, and then simplifying the order of the system such that a sequence of lower-fidelity models is formed. Due to the criticality of the assumptions between each level of fidelity, this process of model reduction is application-specific and discipline-specific; thus requiring a substantial amount of insight into the physics behind the problem. More on the model reduction process and a homotopy path to problem formulation can be found in Ref. [154]. The next section gives an example of a homotopy path to formulating a 6-DOF flight vehicle dynamical model.

3. Example Homotopy Path to High-Fidelity Flight Vehicle Model

The following development illustrates a homotopy modeling sequence from low-order (2-DOF) to high-order (6-DOF) flight vehicle equations of motion. These problems, P1 to P8, assume a flat-earth, small roll/bank angles, and coordinated turns$(\beta = 0)$; however, the same process can include spherical earth terms, large angle aerodynamic relations, and nonzero sideslip.

2-DOF

$$\text{P1}: \quad \underline{x} = [x, z, V, \gamma]^T \in \mathbb{R}^4$$
$$\underline{u} = [\alpha] \in \mathbb{R}^1$$

$$\dot{x} = V \cos\gamma$$
$$\dot{z} = V \sin\gamma$$
$$\dot{V} = \frac{-\mathbf{D}(\underline{x}, \underline{u})}{m} - g(z)\sin\gamma$$
$$\dot{\gamma} = \frac{\mathbf{L}(\underline{x}, \underline{u})}{mV} - \frac{g(z)}{V}\cos\gamma$$

$$\text{P2}: \quad \underline{x} = [x, z, V, \gamma, \alpha]^T \in \mathbb{R}^5$$
$$\underline{u} = [u_\alpha] \in \mathbb{R}^1$$

$$\dot{x} = V \cos\gamma$$
$$\dot{z} = V \sin\gamma$$
$$\dot{V} = \frac{-\mathbf{D}(\underline{x})}{m} - g(z)\sin\gamma$$
$$\dot{\gamma} = \frac{\mathbf{L}(\underline{x})}{mV} - \frac{g(z)}{V}\cos\gamma$$
$$\dot{\alpha} = u_\alpha$$

3-DOF

$$\text{P3}: \quad \underline{x} = [x, y, z, V, \gamma, \xi]^T \in \mathbb{R}^6$$
$$\underline{u} = [\alpha, \sigma] \in \mathbb{R}^2$$

$$\dot{x} = V \cos\gamma \cos\xi$$
$$\dot{y} = V \cos\gamma \sin\xi$$
$$\dot{z} = V \sin\gamma$$
$$\dot{V} = \frac{-\mathbf{D}(\underline{x}, \underline{u})}{m} - g(z)\sin\gamma$$
$$\dot{\gamma} = \frac{\mathbf{L}(\underline{x}, \underline{u})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma$$
$$\dot{\xi} = \frac{\mathbf{L}(\underline{x}, \underline{u})\sin\sigma}{mV\cos\gamma}$$

$$\text{P4}: \quad \underline{x} = [x, y, z, V, \gamma, \xi, \alpha, \sigma]^T \in \mathbb{R}^8$$
$$\underline{u} = [u_\alpha, u_\sigma] \in \mathbb{R}^2$$

$$\dot{x} = V \cos\gamma \cos\xi$$
$$\dot{y} = V \cos\gamma \sin\xi$$
$$\dot{z} = V \sin\gamma$$
$$\dot{V} = \frac{-\mathbf{D}(\underline{x})}{m} - g(z)\sin\gamma$$
$$\dot{\gamma} = \frac{\mathbf{L}(\underline{x})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma$$
$$\dot{\xi} = \frac{\mathbf{L}(\underline{x})\sin\sigma}{mV\cos\gamma}$$
$$\dot{\alpha} = u_\alpha$$
$$\dot{\sigma} = u_\sigma$$

4~5-DOF

P5: $\underline{x} = [x, y, z, V, \gamma, \xi, \alpha, \sigma]^T \in \mathbb{R}^8$

$\underline{u} = [p, q, r] \in \mathbb{R}^3$

$$\dot{x} = V \cos\gamma \cos\xi$$
$$\dot{y} = V \cos\gamma \sin\xi$$
$$\dot{z} = V \sin\gamma$$
$$\dot{V} = \frac{-\mathbf{D}(\underline{x})}{m} - g(z)\sin\gamma$$
$$\dot{\gamma} = \frac{\mathbf{L}(\underline{x})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma$$
$$\dot{\xi} = \frac{\mathbf{L}(\underline{x})\sin\sigma}{mV\cos\gamma}$$
$$\dot{\alpha} = q - \left(\frac{\mathbf{L}(\underline{x})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma\right)$$
$$\dot{\sigma} = \frac{\mathbf{L}(\underline{x})\sin\sigma}{mV}\tan\gamma + p\cos\alpha + r\sin\alpha$$

P6: $\underline{x} = [x, y, z, V, \gamma, \xi, \alpha, \sigma, p, q, r]^T \in \mathbb{R}^{11}$

$\underline{u} = [u_p, u_q, u_r] \in \mathbb{R}^3$

$$\dot{x} = V \cos\gamma \cos\xi$$
$$\dot{y} = V \cos\gamma \sin\xi$$
$$\dot{z} = V \sin\gamma$$
$$\dot{V} = \frac{-\mathbf{D}(\underline{x})}{m} - g(z)\sin\gamma$$
$$\dot{\gamma} = \frac{\mathbf{L}(\underline{x})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma$$
$$\dot{\xi} = \frac{\mathbf{L}(\underline{x})\sin\sigma}{mV\cos\gamma}$$
$$\dot{\alpha} = q - \left(\frac{\mathbf{L}(\underline{x})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma\right)$$
$$\dot{\sigma} = \frac{\mathbf{L}(\underline{x})\sin\sigma}{mV}\tan\gamma + p\cos\alpha + r\sin\alpha$$
$$\dot{p} = u_p$$
$$\dot{q} = u_q$$
$$\dot{r} = u_r$$

6-DOF

P7:

$\underline{x} = [x, y, z, V, \gamma, \xi, \alpha, \sigma, p, q, r]^T \in \mathbb{R}^{11}$

$\underline{u} = [\underline{\delta}] \in \mathbb{R}^8$

$$\dot{x} = V \cos\gamma \cos\xi$$

$$\dot{y} = V \cos\gamma \sin\xi$$

$$\dot{z} = V \sin\gamma$$

$$\dot{V} = \frac{-\mathbf{D}(\underline{x}, \underline{\delta})}{m} - g(z)\sin\gamma$$

$$\dot{\gamma} = \frac{\mathbf{L}(\underline{x}, \underline{\delta})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma$$

$$\dot{\xi} = \frac{\mathbf{L}(\underline{x}, \underline{\delta})\sin\sigma}{mV\cos\gamma}$$

$$\dot{\alpha} = q - \left(\frac{\mathbf{L}(\underline{x}, \underline{\delta})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma \right)$$

$$\dot{\sigma} = \frac{\mathbf{L}(\underline{x}, \underline{\delta})\sin\sigma}{mV}\tan\gamma + p\cos\alpha + r\sin\alpha$$

$$\dot{p} = \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}^{\ 2}}\left[\mathbf{M}_x(\underline{x}, \underline{\delta}) - \left(I_{zz} - I_{yy}\right)qr + I_{xz}qp\right] + \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^{\ 2}}\left[\mathbf{M}_z(\underline{x}, \underline{\delta}) + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right]$$

$$\dot{q} = \frac{1}{I_{yy}}\left[\mathbf{M}_y(\underline{x}, \underline{\delta}) - \left(I_{xx} - I_{zz}\right)pr - I_{xz}\left(p^2 - r^2\right)\right]$$

$$\dot{r} = \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^{\ 2}}\left[\mathbf{M}_x(\underline{x}, \underline{\delta}) + \left(I_{yy} - I_{zz}\right)qr + I_{xz}qp\right] + \frac{I_x}{I_{xx}I_{zz} - I_{xz}^{\ 2}}\left[\mathbf{M}_z(\underline{x}, \underline{\delta}) + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right]$$

P8:

$$\underline{x} = [x, y, z, V, \gamma, \xi, \alpha, \sigma, p, q, r, \delta_1, ..., \delta_8]^T \in \mathbb{R}^{19}$$

$$\underline{u} = [u_{\underline{\delta}}] \in \mathbb{R}^8$$

$$\dot{x} = V \cos\gamma \cos\xi$$

$$\dot{y} = V \cos\gamma \sin\xi$$

$$\dot{z} = V \sin\gamma$$

$$\dot{V} = \frac{-\mathbf{D}(\underline{x})}{m} - g(z)\sin\gamma$$

$$\dot{\gamma} = \frac{\mathbf{L}(\underline{x})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma$$

$$\dot{\xi} = \frac{\mathbf{L}(\underline{x})\sin\sigma}{mV\cos\gamma}$$

$$\dot{\alpha} = q - \left(\frac{\mathbf{L}(\underline{x})\cos\sigma}{mV} - \frac{g(z)}{V}\cos\gamma \right)$$

$$\dot{\sigma} = \frac{\mathbf{L}(\underline{x})\sin\sigma}{mV}\tan\gamma + p\cos\alpha + r\sin\alpha$$

$$\dot{p} = \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}^2}\left[\mathbf{M}_x(\underline{x}) - \left(I_{zz} - I_{yy}\right)qr + I_{xz}qp\right] + \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2}\left[\mathbf{M}_z(\underline{x}) + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right]$$

$$\dot{q} = \frac{1}{I_{yy}}\left[\mathbf{M}_y(\underline{x}) - \left(I_{xx} - I_{zz}\right)pr - I_{xz}\left(p^2 - r^2\right)\right]$$

$$\dot{r} = \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2}\left[\mathbf{M}_x(\underline{x}) + \left(I_{yy} - I_{zz}\right)qr + I_{xz}qp\right] + \frac{I_{x}}{I_{xx}I_{zz} - I_{xz}^2}\left[\mathbf{M}_z(\underline{x}) + \left(I_{xx} - I_{yy}\right)qp - I_{xz}qr\right]$$

$$\dot{\delta}_i = u_{\delta_i} \quad i = 1, ..., 8$$

These equations can be modified to include sideslip by adding the corresponding sideslip terms in the dynamical equations and adding β as a control in P3, a pseudo-control in P4, etc. until the inclusion of $\dot{\beta}$ given by

$$\dot{\beta} = \left(-\frac{\mathbf{Y}(\underline{x})}{mV} + g(z)\cos\gamma\sin\sigma \right) + p\sin\alpha - r\cos\alpha \qquad (3.113)$$

H. CONCLUDING REMARKS

This chapter's main purpose was to characterize the vehicle and to summarize the development of the 6-DOF model and its environment for use in the high-fidelity trajectory optimization simulations presented in Chapter VI. Since all of the work presented in this dissertation was based on the X-33 RLV, an overview description was necessary to familiarize the reader with the size, functionality, and general characteristics of the vehicle used in the trajectory optimization simulations that follow. Although the parameters are specific to the X-33 in this work, the guidance and control method is vehicle-independent and as such, the data for any RLV, or flight vehicle for that matter, could be substituted. As presented, a 3-DOF model is readily available from simplification of the full, 6-DOF equations and will be applied in the next chapter. In addition, as with all engineering efforts involving the modeling and simulation of complex systems, an ideal hierarchy to increasing fidelity exists; however, this path is not always clear. The presented homotopy path to modeling is just one of many ways to systematically solve complex problems.

IV. 3-DOF REAL-TIME TRAJECTORY OPTIMIZATION

A. INTRODUCTION

Historically, trajectory design (e.g.,, path planning) and analysis was a tedious, complex process conducted offline that often took days, weeks, or months. Recently, with new techniques and more powerful computer technology, trajectory generation is a lot simpler and can be done online in seconds, minutes, or hours. In recent years, it has been shown [83],[84] that pseudospectral (PS) methods are capable of generating real-time trajectories for a 3-DOF dynamical model for the Space Shuttle vehicle model parameters. Real-time and near real-time trajectory generation is now possible. Not only is it possible, but the laborious design process has been streamlined to simply setting up the problem in terms of model parameters, state variables, boundary conditions and constraints, i.e., the optimal control problem formulation, and then using a optimization tool to solve. Using such formulation for a 3-DOF dynamical model and a PS-based optimization method, this chapter provides optimal open-loop and closed-loop solutions with three primary purposes: 1.) to combine and extend emerging concepts in real-time optimal control to address the difficulties in entry guidance, 2.) to demonstrate the feasibility of using such a unified approach based on optimal control theory as a highly effective and general-purpose guidance method, and 3.) to investigate various ideas using the 3-DOF model that can be extended to the full, 6-DOF methodology; hence, serving as a framework or baseline.

Since recent work has employed the use of optimal control via DIDO to address specific problems like range maximization [83], trajectory reshaping with effector failures [46], and footprint determination [47],[86], it is the intent of this work to combine these contributions with many of the emerging concepts in real-time optimal control to address the difficulties in entry guidance. To this end, it circumvents the traditional problems that arise in entry segmentation, such as the concept of Terminal Area Energy Management (TAEM), and instead embarks on using a single algorithm to guide the vehicle from its entry conditions all the way to a neighborhood of the landing site. Thus, given the coordinates of a landing site, the goal of the entry guidance algorithm is to guide the vehicle to an automatically generated final approach corridor for handover to an autoland program. In principle, it is possible for a PS-based method to perform autoland as well, but for the purpose of limiting the scope of this research, autoland guidance is omitted. In addition to developing an automated method for generating terminal guidance conditions and as a result of using nonlinear optimal control and the elimination of trajectory segmentation, the need for gain scheduling is also removed. This provides a unique and effective approach facilitating an on-board, autonomous capability.

To demonstrate how using nonlinear trajectory optimization to solve the RLV reentry problem can provide on-board, autonomous capability, this chapter entails open-loop solutions used to generate landing footprints and intelligent path planning to terminal guidance conditions with an example illustrating a feasibility-based landing site re-targeting scenario. Likewise, this chapter goes beyond previous work by implementing a PS-feedback method to generate closed-loop, optimal guidance that compensates for large uncertainties and disturbances.

Based on the possibility of real-time optimal control, a PS feedback theory has been formulated by Ross et al. [148]. For entry guidance, this approach updates the guidance laws in the so-called Carathèodory sense, rather than the standard sampled data feedback approach. This type of an update permits a longer computational time that may be exploited to solve high-fidelity trajectory optimization problems. An introduction to this approach is described in Chapter II along with details on the PS method and the spectral algorithm (packaged into DIDO) used to generate the optimal controls. An important result of the closed-loop work is the revelation that under the PS guidance law, the simplified X-33-based generic vehicle is capable of guiding itself to a landing site despite the extreme situation of hurricane wind effects that exceed Category 5.

Overall, the 3-DOF work presented in this work is intended to serve as a building block to the full, 6-DOF reentry problem. Following suite as in most engineering practices, it is advantageous to test the various guidance and control ideas on a simpler system or a reduced-order model before proceeding to more complex, higher-order systems.

B. OPEN-LOOP, OPTIMAL TRAJECTORY GENERATION

1. 3-DOF Model

To confirm the feasibility of this method for solving the RLV reentry problem, various open-loop problems were formulated and solved. For all of the results presented in this section, the equations of motion used in the optimal control formulation are the standard 3-DOF dynamical model for an entry vehicle in a rotating atmosphere around an inverse-square law gravitational field as explained in Chapter III and repeated here for completeness.

$$
\begin{aligned}
\dot{r} &= V\sin\gamma \\
\dot{\mu} &= \frac{V\cos\gamma\cos\xi}{r\cos\lambda} \\
\dot{\lambda} &= \frac{V\cos\gamma\sin\xi}{r} \\
\dot{V} &= -\frac{D(\alpha,V,r)}{m} - g(r)\sin\gamma + \Omega^2 r\cos\lambda(\sin\gamma\cos\lambda - \cos\gamma\sin\lambda\sin\xi) \\
\dot{\gamma} &= \frac{L(\alpha,V,r)\cos\sigma}{mV} + \left(\frac{V}{r} - \frac{g(r)}{V}\right)\cos\gamma \\
&\quad + 2\Omega\cos\lambda\cos\xi + \frac{\Omega^2 r}{V}\cos\lambda(\cos\gamma\cos\lambda + \sin\gamma\sin\lambda\sin\xi) \\
\dot{\xi} &= \frac{L(\alpha,V,r)\sin\sigma}{mV\cos\gamma} - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda \\
&\quad + 2\Omega(\tan\gamma\cos\lambda\sin\xi - \sin\lambda) - \frac{\Omega^2 r}{V\cos\gamma}\sin\lambda\cos\lambda\cos\xi
\end{aligned}
\tag{4.1}
$$

where $g = GM/r^2$ is the inverse-square gravitational acceleration, $r = R_E + h$ is the radial position measured from the center of the Earth, μ is the geocentric longitude, λ is the geocentric latitude, V is the total airspeed (i.e., velocity magnitude), γ is the vehicle's flight path angle (FPA), and ξ is the vehicle's heading angle. The lift and drag forces are represented as L and D, respectively, given by

$$L = \frac{1}{2}\rho(r)V^2 C_L(\alpha, M(V,r))S_{ref} = \frac{1}{2}(\rho_0 e^{-\beta(r-r_0)})V^2 C_L(\alpha, M(V,r))S_{ref} \tag{4.2}$$

$$D = \frac{1}{2}\rho(r)V^2 C_D(\alpha, M(V,r))S_{ref} = \frac{1}{2}(\rho_0 e^{-\beta(r-r_0)})V^2 C_D(\alpha, M(V,r))S_{ref} \tag{4.3}$$

where the term $\dfrac{1}{2}\rho(r)V^2$ is the dynamic pressure, $\overline{q}$, and the term $\rho_0 e^{-\beta(r-r_0)}$ is a two-parameter model for atmospheric density, ρ, as a function of scaled altitude.

The aerodynamic coefficients of lift (C_L) and drag (C_D) are functions of Mach number and angle-of-attack (AoA) that are approximated by fitting X-33 data with 2nd-order polynomials

$$C_L(\alpha, M(V,r)) = -0.0005225\alpha^2 + 0.03506\alpha - 0.04857M + 0.1577 \quad (4.4)$$

$$C_D(\alpha, M(V,r)) = 0.0001432\alpha^2 + 0.00558\alpha - 0.01048M + 0.2204 \quad (4.5)$$

where the Mach number, $M = V / a(r)$, is explicitly dependent on velocity and speed-of-sound, *a*, at a specific altitude. Figure 4.1 and Figure 4.2 show the 3D-surface plots of the actual lift and drag data with their corresponding polynomial surface-fits respectively. These particular poly-fits were generated using the online tool *ZunZun*©, as described in Appendix A. The fitting algorithm is based on a least-squares error method that allows the user to select specific fitting functions (e.g.,, polynomial, nonlinear, etc.) and/or parameters like the order of the polynomial, etc. In addition to allowing users to visually inspect the fits and the statistics, it ranks numerous fits based on a user specified statistical metric like standard deviation. After selecting a few polynomials of varying order from *ZunZun*, rudimentary plotting tests were conducted to compare their accuracy for various fixed Mach numbers over the entire alpha-range. Although other functions resulted in better fits, the accuracy of the 2nd-order poly-fit was deemed satisfactory for *M*=1~6. Also, its simplicity from an implementation and numerical perspective made it a logical choice for this “proof-of-concept” study.

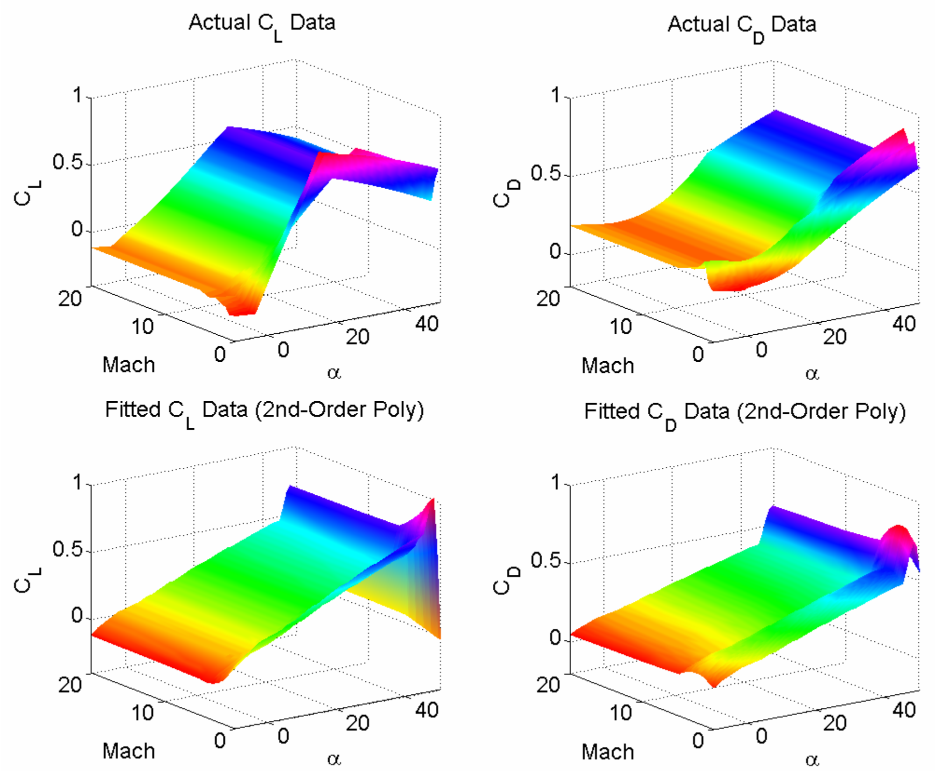


Figure 4.1 Lift and Drag Coefficients: Actual and Fitted Surface Plots.

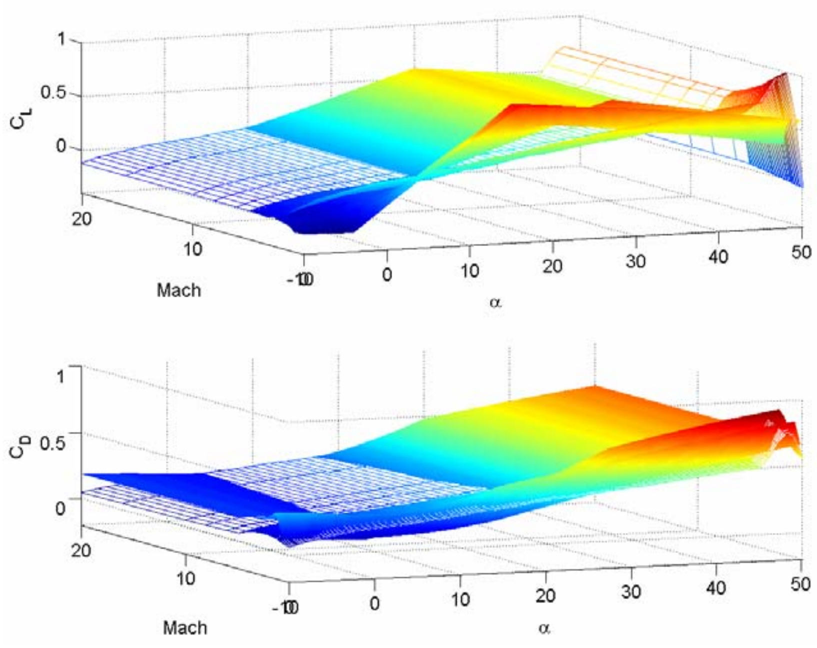


Figure 4.2 Comparison of Actual and Fitted Surface Plots.

Note that a description of the tools used for the surface fits of the X-33's large tabulated aero-data with additional levels of accuracy can be found in Appendix A. The additional parameters in Eq. (4.1) are defined in Table 4.1.

Table 4.1 Model Parameters

Symbol	Quantity	Value/Unit
m	RLV Empty Mass	2455 slugs
S_{ref}	RLV Ref. Area	1608 ft^2
ρ_0	Standard Density	0.002378 slugs/ft^3
r_0	Ref. Altitude	20902900 ft
β	Inverse Scale Height	4.20168e-5 ft^{-1}
GM	Earth's Gravitational Constant	0.14076539e17 ft^3/s^2
Ω	Earth's Angular Velocity	7.2722e-5 rad/s
R_e	Earth's Radius	20925646.32 ft

In accordance with the general nonlinear system, this section provided a mathematical model for the dynamical system. With the dynamical model now defined, the problem can next be formulated in terms of the nonlinear optimal control problem (NL OCP) presented in Chapter. II.

2. Problem Formulation

Consistent with the NL OCP formulation, this section develops the state and control vectors, cost function, and remaining constraints.

a. General Assumptions

The overall objective of the reentry problem posed as an OCP is to find an optimal flight trajectory from LEO (or a suborbital entry altitude), through the atmosphere, to some intended landing site while minimizing some performance index and without violating any path constraints such as thermal or structural limits. The three general assumptions for all problem formulations herein are that the RLV is (1) a rigid-body, (2) a lifting-body, and (3) unpowered.

First, a rigid-body assumption eliminates the distortional effects caused by a flexible body, such as elastic degrees-of-freedom needed to describe the vehicle's motion and additional forces and moments due to aeroelasticity that affects structural bending, torsion, and flow properties (e.g.,, flutter). Accounting for distortion usually involves alterations to the aerodynamic derivatives such as adding multiplicative flex/rigid ratios to account for the effects of static bending. For trajectory optimization, these effects are negligible.

Second, the RLV's body is considered to contribute substantially to the overall lift forces generated during reentry. Variations in aerodynamic forces affect the control of a descent trajectory for a lift-producing vehicle as opposed to a strictly ballistic vehicle.

Third, typical RLVs re-enter the atmosphere under no power. Although some vehicles use Reaction Control System (RCS) thrusters for maneuvering at high altitudes where control surfaces are ineffective, this RLV reentry problem will assume no thrust and is therefore strictly a gliding descent trajectory.

b. State and Control Vectors

For this problem, the state variables consist of geographical position and velocity terms that together form a six-dimensional state space given by

$$\mathbf{X} = \{(r, \mu, \lambda, V, \gamma, \xi) \in \mathbb{R}^6 : \; x_i^L \leq x_{i=1..6} \leq x_i^U\} \tag{4.6}$$

where the corresponding bounds for each state variable are

$$\begin{bmatrix} 0\,\text{ft} \\ -90\,\text{deg} \\ -89\,\text{deg} \\ 1.0\,\text{ft/s} \\ -89\,\text{deg} \\ -180\,\text{deg} \end{bmatrix} \leq \begin{bmatrix} h \\ \mu \\ \lambda \\ V \\ \gamma \\ \xi \end{bmatrix} \leq \begin{bmatrix} 400000\,\text{ft} \\ 90\,\text{deg} \\ 89\,\text{deg} \\ 17060\,\text{ft/s} \\ 89\,\text{deg} \\ 180\,\text{deg} \end{bmatrix} \tag{4.7}$$

Likewise, the control variables are first selected based on the assumption of no command delays and are represented by the physical modulation of the vehicle's AoA (α) and bank angle (σ) such that the control space is given by

$$\mathbf{U} = \{(\alpha, \sigma) \in \mathbb{R}^2 : u_i^L \leq u_{i=1..6} \leq u_i^U\} \tag{4.8}$$

and the bounds for each control variable are

$$\begin{bmatrix} -10\,\text{deg} \\ -80\,\text{deg} \end{bmatrix} \leq \begin{bmatrix} \alpha \\ \sigma \end{bmatrix} \leq \begin{bmatrix} 50\,\text{deg} \\ 80\,\text{deg} \end{bmatrix} \tag{4.9}$$

Without a thrust force for the reentry gliding problem, the only controllable parameters for the 3-DOF model are the lift and drag forces. Typically, for symmetric flight (i.e., coordinated turns with zero sideslip angle, β , the lift and drag coefficients can be determined by the vehicle's AoA and Mach number, a function of velocity and speed-of-sound at a given altitude. However, it is the physical modulation of AoA and bank angle (BA) that controls the vehicle's translational motion through the atmosphere. Therefore, a common control vector for the reentry problem is $\mathbf{u} = [\alpha, \sigma]^T \in \mathbb{R}^2$ as given in Eq. (4.8) Of course, this control vector assumes that there are no command delays (i.e., lags) and is sometimes referred to as "inertialess" control. To add more realism to the problem, as explained in Refs. [46] and [47], the rate limits are modeled by forming a new "virtual" control vector mathematically expressed as

$$\mathbf{u} = [u_\alpha \;\; u_\sigma]^T \in \mathbb{R}^2 \tag{4.10}$$

Now, the original state vector must include the physical controls, α and σ, to form a new state vector

$$\mathbf{x} = [r \;\mu\; \lambda\; V\; \gamma\; \xi\; \alpha\; \sigma]^T \in \mathbb{R}^8 \tag{4.11}$$

Remark: For a real vehicle, or in a 6-DOF simulation, it is the control surface deflections that create body moments and forces to augment the wind-relative AoA, BA, and sideslip angle. This is addressed by the 6-DOF models of Chapter VI.

c. Cost and Endpoint Conditions

Even from a sub-orbital entry altitude, the RLV must traverse a fairly large portion of the Earth's surface, descend through the Earth's atmosphere, and reach a designated landing site with the appropriate velocity and attitude. In principle, it is possible to use the entry guidance algorithm developed in this work for autolanding as well; however, since the present model does not take into account aerodynamic ground effects, efforts are focused in guiding the RLV all the way up to a handover to autoland guidance. As such, this formulation discards the traditional notion of segmented guidance schemes that involve such concepts as TAEM; instead, a Final Approach Corridor (FAC), as shown in Figure 4.3, is designed that provides the capability to automatically generate final endpoint conditions.

(1) Design of the Final Approach Corridor. Recent efforts have considered other means of implementing the transition from TAEM to A/L guidance. For example, Ref. [66] employs terminal conditions based on a TAEM "box" specified by altitude, range, and heading that exists as a latitude/longitude target that is tangent to the HAC. However, in a guidance scheme developed around optimal control theory that specifies the final desired conditions, the additional geometry of the HAC is unnecessary. Simply specifying the desired vehicle attitude, heading, and velocity at the onset of the A/L phase along with the inclusion of the other constraints such as dynamic pressure, heat, and normal load, the guidance will automatically determine the necessary maneuvers. As such, the following FAC-target does not use the HAC.

The basic premise of the FAC-target design is to project all the final constraints onto a specific three-dimensional "box" based on any desired landing location (i.e., runway). Figure 4.3 illustrates how the "box" geometry is extracted from the FAC. For a simplified terminal guidance reentry problem posed in the typical TPBV fashion, position (i.e., latitude, longitude, and altitude) and velocity are usually specified as the final conditions. Since the pre-A/L phase of flight (i.e., approach setup) is critical in a practical landing scenario, it is important to include the vehicle's attitude and velocity (both magnitude and vertical component) in the specified final conditions. Note that these values can vary depending on the vehicle characteristics and the desired runway.

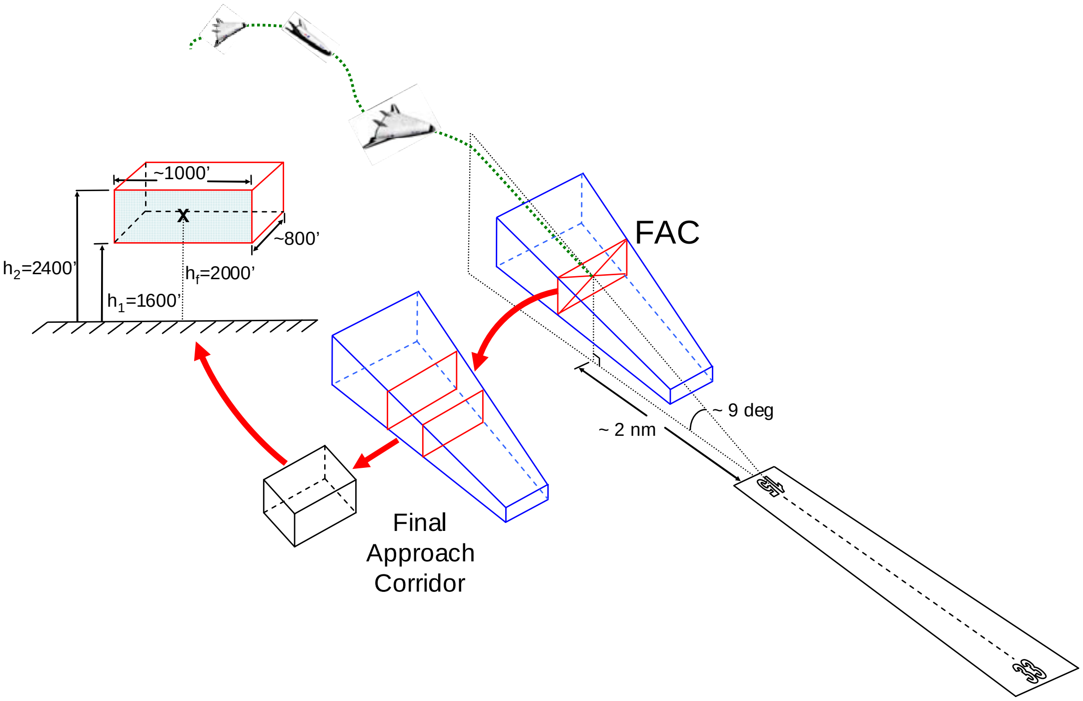


Figure 4.3 Final Approach Corridor (FAC) for "Target" Conditions.

For this research work, an algorithm was designed that automatically calculates the FAC-target for a landing at any Florida runway that's length is greater than or equal to 10,000 ft. The NASA Shuttle Landing Facility (KTTS) in Titusville, FL was the primary landing site and served as the example for developing the FAC-generation logic. Given the runway's center location (latitude = 28.615 deg, longitude = -80.6945 deg) and its heading with respect to magnetic North (150/330 deg), the center of the FAC is generated and used to determine the respective geometry for the remainder of the FAC-cube. Figure 4.4 gives an example of how the coordinates A, B, and C are determined simply by trigonometric relations based soley on the runway's position and orientation. In a similar fashion, the remaining coordinates are determined for the 3D-cube and then mapped to a constraint set. For the demonstation purpose of this research, the mapped constraints were only "box-constraints." A more accurate and effective approach is to map the FAC-region as a path constraint similar to the constraint modeling in Ref. [178]. For a simplified case, the center of the FAC alone can be used in DIDO as the final condition point. Also note that the FAC positional information is assumed to be fixed; however, the geometric parameters can be adjusted as desired. For example, increasing the FAC's distance from the runway can allow for a shallower glide slope. Note that the distance from the runway should be measured from the landing threshold (close to the end of the runway) and as such the length of the runway would become a required input parameter in order to shift the measurements from the center of the runway.

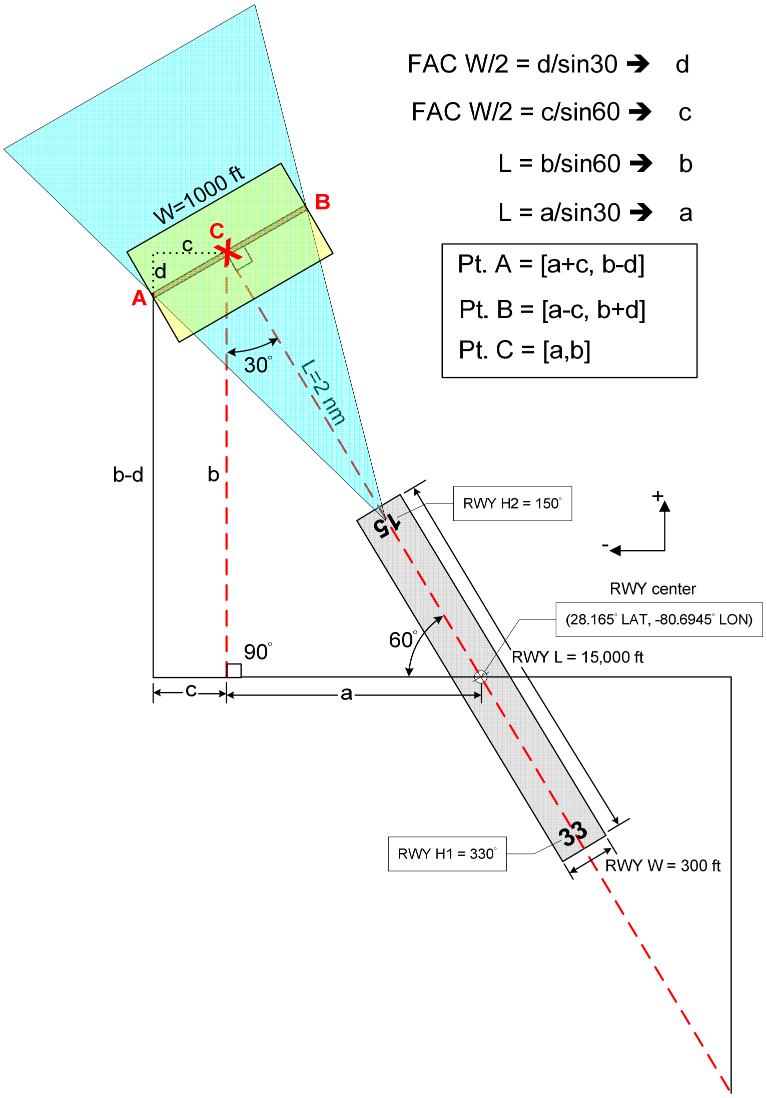


Figure 4.4 Runway Geometry used to Generate FAC.

An advantage to this design is that the entire descent trajectory is based on achieving the final conditions and as such there is no need to switch runway approach at the last minute based on energy condition. Use of the FAC in an optimal guidance and control scheme will ensure that the vehicle arrives with the correct energy and eliminates the need for change in approach. This can potentially improve landing safety and may be worth further investigations via Monte Carlo analysis.

(2) Generating the Cost Function and the Final Conditions. As mentioned, the FAC setup includes the RLV's alignment with the runway centerline at the appropriate altitude, velocity, etc.; hence, the FAC manifests a projection of all the final conditions onto a three-dimensional cube. The problem is then to find the control history that minimizes some cost function associated with intercepting an automatically generated FAC based on a desired landing site. After all, the primary objective of RLV reentry is to safely arrive at a designated landing site with the appropriate flight conditions. Note that the runway geometry for the FAC generation logic is determined from a pre-programmed U.S. database of runway information [179],[180] . In addition to this objective cost, heat, dynamic pressure, structural loads, drag, or some other criterion could be minimized, and these can be modeled as path constraints. The reentry problem is then formulated by appropriately defining the cost function and corresponding constraints.

A general expression for a terminal guidance performance index is defined by the following cost function,

$$J(\underline{x}(\cdot),\underline{u}(\cdot),t_0,t_f)=\sum_{i=1}^{n} w_i\Delta s_i^{\,2}=w_1\Delta s_1^{\,2}+w_2\Delta s_2^{\,2}+\ldots w_n\Delta s_n^{\,2} \qquad (4.12)$$

where $\Delta s_i = \left|s_2 - s_1\right|_i$ is the difference between a specific final state value, $x_i(t_f)$, and a desired final condition (i.e., the "target") for that state, and w_i is the corresponding weighting term that can be adjusted depending on the relative importance of the condition. For the models using the FAC-target in this work, the cost function can simply be the difference between the center of the FAC and the vehicle's final position in coordinates of altitude, latitude, and longitude designated as Δh, $\Delta\lambda$, $\Delta\mu$, respectively and equally weighted $(w_{1,2,3}=1)$ such that

$$J(\cdot)=1\cdot\Delta h^2+1\cdot\Delta\lambda^2+1\cdot\Delta\mu^2=\left(h_{FAC}-h_f\right)^2+\left(\lambda_{FAC}-\lambda_f\right)^2+\left(\mu_{FAC}-\mu_f\right)^2 \qquad (4.13)$$

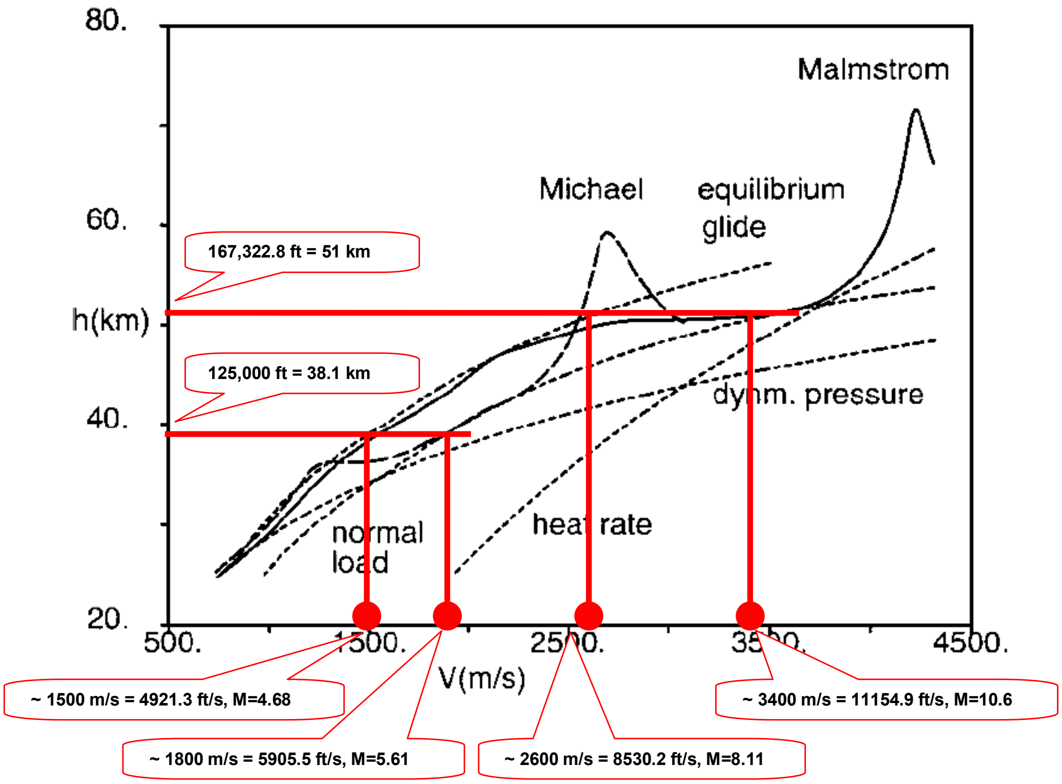
In this context, the OCP is to find the control history that minimizes the miss distance to the center of the FAC with the general endpoint (boundary) conditions

$$\begin{aligned} e(x_0, t_0) &:= \mathbf{x}(t_0) = [h_0, \mu_0, \lambda_0, V_0, \gamma_0, \xi_0]^T \\ e(x_f, t_f) &:= \mathbf{x}(t_f) = [h_f, \mu_f, \lambda_f, V_f, \gamma_f, \xi_f]^T \end{aligned} \tag{4.14}$$

Note that the initial conditions for most of the 3-DOF results are selected based on X-33 reference trajectories generated from previous research work as depicted in Figure 4.5 [29]. When using the FAC for terminal guidance, the final conditions represented as equality constraints in Eq. (4.14) are more appropriately represented as inequality constraints where the FAC-box defines the constraints for h_f, μ_f, and λ_f such as

$$\begin{aligned} \left(\mathrm{h}_{\mathrm{FAC}} - \frac{\mathrm{FAC}_{\mathrm{H}}}{2}\right) &\leq h_f \leq \left(\mathrm{h}_{\mathrm{FAC}} + \frac{\mathrm{FAC}_{\mathrm{H}}}{2}\right) \\ \left(\mu_{\mathrm{FAC}} - \frac{\mathrm{FAC}_{\mathrm{W}}}{2}\right) &\leq \mu_f \leq \left(\mu_{\mathrm{FAC}} + \frac{\mathrm{FAC}_{\mathrm{W}}}{2}\right) \\ \left(\lambda_{\mathrm{FAC}} - \frac{\mathrm{FAC}_{\mathrm{L}}}{2}\right) &\leq \lambda_f \leq \left(\lambda_{\mathrm{FAC}} + \frac{\mathrm{FAC}_{\mathrm{L}}}{2}\right) \end{aligned} \tag{4.15}$$

where L = length, W = width, and H = height of the 3D FAC-box.



167,322.8 ft = 51 km

125,000 ft = 38.1 km

~ 1500 m/s = 4921.3 ft/s, M=4.68

~ 1800 m/s = 5905.5 ft/s, M=5.61

~ 2600 m/s = 8530.2 ft/s, M=8.11

~ 3400 m/s = 11154.9 ft/s, M=10.6

Figure 4.5 Selection of Various Initial Conditions from X-33 Reference Trajectories from [29].

When the problem is to maximize downrange (μ) and/or crossrange (λ), such as the case for generating landing footprints, the performance index is defined by the following cost functions,

$$Min\,\{\,J(\cdot)\ =-\mu_f\}\qquad -\text{or}-\qquad Min\,\{\,J(\cdot)\ =\pm\lambda_f\} \tag{4.16}$$

The results in this chapter consist of both cost functions, Eqs. (4.13) and (4.16).

d. Path Constraints

In addition to the constraints on dynamics, states, controls, and endpoints (i.e., boundary conditions), a critical set of constraints for the reentry problem that cannot be neglected are those associated with structural loads (e.g.,, normal acceleration), dynamic pressure, and heat rate given as,

$$\begin{aligned}
&n_z(r,V,\alpha) = L(r,V,\alpha)\cos\alpha + D(r,V,\alpha)\sin\alpha \\
&\bar{q}(r,V) = \frac{1}{2}\rho(r)V^2 \\
&Q(r,V) = k\sqrt{\rho(r)}\,V^{3.15}\,;\ k = 4.47228e-9
\end{aligned} \tag{4.17}$$

respectively. Thus, the following path constraints where used throughout this work

$$\begin{bmatrix} -2.5\text{g's} \\ 0 \\ 0 \end{bmatrix} \le \begin{bmatrix} n_z(r,V,\alpha) \\ \bar{q}(r,V) \\ Q(r,V) \end{bmatrix} \le \begin{bmatrix} 2.5\text{g's} \\ 300\,\text{lb/ft}^2 \\ 70\,\text{BTU/ft-s} \end{bmatrix} \tag{4.18}$$

Although the limits for the constraints are vehicle specific, those given by Eq. (4.18) are representative of RLVs like the X-33, X-37, and X-40.

Now, consistent with Eq. (2.1), the specific optimal control formulation for this RLV reentry problem is stated as follows: Given an initial position $(r_0,\ \mu_0,\ \lambda_0)$, velocity magnitude (V_0), FPA (γ_0), heading angle (ξ_0), AoA (α_0), and BA (σ_0), find the control history (α,σ) that minimizes $J(\cdot)$, the miss distance to a predefined target or the max/min range, and subject to the dynamic constraints, Eq. (4.1), the initial and final endpoint conditions, Eq. (4.14), the state constraints, Eq. (4.7), the control constraints, Eq. (4.9), and the path constraints, Eq. (4.18).

3. Solving the Optimal Control Problem

Before numerically solving the problem and analyzing results, it is first necessary to apply optimal control theory and develop the key necessary optimality conditions for the given problem. The proceeding theoretical results may aid in the verification of the numerical solution as presented in the next section. The development of the necessary conditions follows the procedures as outlined in Chapter II. As such, the first step requires the formulation of the Hamiltonian:

$$H(\underline{P},\underline{x},\underline{u},t) = F(\underline{x},\underline{u},t) + P^T f(\underline{x},\underline{u},t) \tag{4.19}$$

where $F(\cdot)$ is the Lagrange cost, P represents a vector of Lagrange multipliers, and $f(\cdot)$ is the vector field for the right hand side of the differential equations. With

$$\underline{P} = [P_r, P_\mu, P_\lambda, P_V, P_\gamma, P_\xi]^T \in \mathbb{R}^6\,; \qquad \underline{x}_f = [r_f, \mu_{f,} \lambda_f, V_f, \gamma_f, \xi_f] \in \mathbb{R}^6 \tag{4.20}$$

and omitting the $\Omega-$terms for brevity, the Hamiltonian becomes

$$\begin{aligned}H(\cdot) = P_r\left[V\sin\gamma\right] + P_\mu\left[\frac{V\cos\gamma\cos\xi}{r\cos\lambda}\right] + P_\lambda\left[\frac{V\cos\gamma\sin\xi}{r}\right]\ldots \\ +P_V\left[-\frac{1}{2m}\rho_0 e^{-\beta(r-r_0)}V^2 C_D(\alpha,M)S_{ref} - g(r)\sin\gamma + \ldots\right]\ldots \\ +P_\gamma\left[\frac{1}{2m}\rho_0 e^{-\beta(r-r_0)}VC_L(\alpha,M)S_{ref}\cos\sigma + \left(\frac{V}{r}-\frac{g(r)}{V}\right)\cos\gamma + \ldots\right]\ldots \\ +P_\xi\left[\frac{1}{2m}\frac{\rho_0 e^{-\beta(r-r_0)}VC_L(\alpha,M)S_{ref}\sin\sigma}{\cos\gamma} - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda + \ldots\right]\end{aligned} \tag{4.21}$$

$$\underline{\dot{x}}(\cdot) = \frac{dH}{d\underline{P}} \Rightarrow RHS\ of\ dynamic\ equations \tag{4.22}$$

Note that the notation used here and in the proceeding chapter differ from that presented in Chapter II due to the selection of state variable notation for longitude (μ) and latitude (λ), corresponding to the dual controls (control covector) and the dual states (costates), respectively.

The Hamiltonian Minimization Condition (HMC) is based on Pontryagin's Minimum Principle such that the optimal control must minimize the Hamiltonian with respect to control. Since the control is subject to an inequality constraint for this problem, the Karush-Kuhn-Tucker (KKT) Theorem is applied by first taking the Lagrangian of the Hamiltonian:

$$\bar{H}(\underline{P},\underline{q},\underline{x},\underline{u},t) = H(\underline{P},\underline{x},\underline{u},t) + \underline{q}^T\underline{h} \tag{4.23}$$

where $\underline{q}$ is a KKT multiplier and $\underline{h}$ is the control and path constraint vector, $\underline{h} := \left[\alpha,\sigma,n_z,\bar{q},Q\right]$. The appropriate necessary condition is then:

$$\frac{\partial\bar{H}}{\partial\underline{u}} = \frac{\partial H}{\partial\underline{u}} + \left(\frac{\partial\underline{h}}{\partial\underline{u}}\right)^T\underline{q} = 0 \tag{4.24}$$

and substituting Eqs. (4.21), (4.4), (4.5), the controls, and the path functions into (4.24), the HMC becomes

$$\begin{cases} P_V\left[-\rho_0 e^{-\beta(r-r_0)}V^2(0.0002\alpha+0.006)S_{ref}/2m\right] \\ \quad +P_\gamma\left[\rho_0 e^{-\beta(r-r_0)}V(-0.0010\alpha+0.035)S_{ref}\cos\sigma/2m\right] \\ \quad +P_\xi\left[\rho_0 e^{-\beta(r-r_0)}V(-0.0010\alpha+0.035)S_{ref}\sin\sigma/2m\cos\gamma\right] \\ \quad +q_\alpha+q_{n_z}\left[\rho_0 e^{-\beta(r-r_0)}V^2S_{ref}/2\right]\begin{bmatrix}(-0.0010\alpha+0.035)\cos\alpha \\ -C_L(\alpha,M)\sin\alpha \\ (0.0002\alpha+0.006)\sin\alpha \\ C_D(\alpha,M)\cos\alpha\end{bmatrix}^T=0 \\ \\ P_\gamma\left[-\rho_0 e^{-\beta(r-r_0)}V(-0.0005\alpha^2+0.035\alpha-0.049M+0.16)S_{ref}\sin\sigma/2m\right] \\ \quad +P_\xi\left[\rho_0 e^{-\beta(r-r_0)}V(-0.0005\alpha^2+0.035\alpha-0.049M+0.16)S_{ref}\cos\sigma/2m\cos\gamma\right] \\ \quad +q_\sigma=0 \end{cases} \tag{4.25}$$

Also, the multiplier-constraint pair must satisfy the following KKT Complementarity Conditions (CC):

$$\begin{gathered} q_{\alpha,\sigma}\begin{cases}\leq 0 & \alpha,\sigma=-10,-80 \\ =0 \quad \mathit{for} & (-10,-80)<\alpha,\sigma<(50,80) \\ \geq 0 & \alpha,\sigma=50,80\end{cases} \\ q_{n_z}\begin{cases}\leq 0 & n_z=-2.5 \\ =0 \quad \mathit{for} & -2.5<n_z<2.5 \\ \geq 0 & n_z=2.5\end{cases} \\ q_{\bar{q}}\begin{cases}=0 & \bar{q}>0 \\ \leq 0 & \bar{q}=300\end{cases}\mathit{for} \\ q_Q\begin{cases}=0 & Q>0 \\ \leq 0 & Q=70\end{cases}\mathit{for} \end{gathered} \tag{4.26}$$

With the Lagrangian of the Hamiltonian defined, information about how the costates vary over time may be gained from the Adjoint Equation:

$$-\dot{\underline{P}}=\frac{\partial\bar{H}}{\partial\underline{x}} \tag{4.27}$$

From Eq. (4.27), most of the adjoints provide no useful information except for that associated with the downrange distance, μ, such that

$$-\dot{P}_\mu = \frac{\partial \bar{H}}{\partial \mu} = 0 \Rightarrow P_\mu = const. \tag{4.28}$$

To determine the final value of the Hamiltonian, the Endpoint Lagrangian, given as,

$$\begin{aligned} \bar{E}(\upsilon, x_f, t_f) &:= E(x_f, t_f) + \upsilon^T e(x_f, t_f) \\ &\Rightarrow \left(r_{FAC} - r_f\right)^2 + \left(\lambda_{FAC} - \lambda_f\right)^2 + \left(\mu_{FAC} - \mu_f\right)^2 \ldots \\ &\quad + \upsilon_r (r_f - r^f) + \upsilon_\mu (\mu_f - \mu^f) + \upsilon_\lambda (\lambda_f - \lambda^f) \ldots \\ &\quad + \upsilon_V (V_f - V^f) + \upsilon_\gamma (\gamma_f - \gamma^f) + \upsilon_\xi (\xi_f - \xi^f) \end{aligned} \tag{4.29}$$

is substituted into the Hamiltonian Value Condition (HVC):

$$H(t_f) + \frac{\partial \bar{E}}{\partial t_f} = 0 \;\Rightarrow\; H(t_f) = 0 \tag{4.30}$$

This indicates that the final value for the Hamiltonian should be zero for this problem. Also, the Endpoint Lagrangian is used to determine the Terminal Transversality Conditions (TTC) such as

$$\begin{aligned} \underline{P}(t_f) &= \frac{\partial \bar{E}}{\partial \underline{x}_f} \\ &= \begin{bmatrix} P_r(t_f) \\ P_\mu(t_f) \\ P_\lambda(t_f) \\ P_V(t_f) \\ P_\gamma(t_f) \\ P_\xi(t_f) \end{bmatrix} = \begin{bmatrix} -2(r_{FAC} - r_f) + \upsilon_r \\ -2(\mu_{FAC} - \mu_f) + \upsilon_\mu \\ -2(\lambda_{FAC} - \lambda_f) + \upsilon_\lambda \\ \upsilon_V \\ \upsilon_\gamma \\ \upsilon_\xi \end{bmatrix} \end{aligned} \tag{4.31}$$

At this point, the TTC does not provide any useful information about the final value of the costates. In addition, the Hamiltonian Evolution Equation (HEE) is used to indicate the nature of the Hamiltonian with respect to time such that:

$$\frac{\partial H}{\partial t} = 0 \;\Rightarrow\; \frac{dH}{dt} = 0 \tag{4.32}$$

Here, the Hamiltonian is constant with respect to time. Combining HEE (4.32) with HVC (4.30), the Hamiltonian should be zero for all time. Although not shown for every specific problem, the above procedure and analysis is used throughout this study where applicable to verify that the numerical results satisfy the necessary conditions of optimality.

4. Results and Analysis

a. Open-Loop Solution

(1) Maximum Downrange Performance. First, the maximum downrange is computed by minimizing the negative of the final downrange distance as given by Eq. (4.16) and repeated here,

$$Min \;\; J(\cdot) \; = -\mu_f \tag{4.33}$$

For this example, the initial and final conditions were specified as,

$$\begin{aligned} &(t_0, h_0, \mu_0, \lambda_0, V_0, \gamma_0, \xi_0) = (0\,\text{sec}, 125000\,\text{ft}, 0^\circ, 0^\circ, 5413\,\text{ft/s}, -1.3^\circ, 0^\circ) \\ &(h_f, V_f, \gamma_f) = (500\,\text{ft}, 335\,\text{ft/s}, -3.0^\circ) \end{aligned} \tag{4.34}$$

where altitude, velocity, and FPA are required for the final constraint set (i.e fixed variables). The final time, longitude, latitude, and heading are unspecified (i.e., free variables) for the maximum range problem. Assuming no a priori knowledge of the expected results, the following guess was used

$$(t_f, \mu_f, \lambda_f, \gamma_f) = (1000\,\text{sec}, 90^\circ, -35^\circ, -90^\circ) \tag{4.35}$$

As seen by the results, this initial guess is extremely poor. Despite these severely bad guesses, the only noticeable effects are that the bank and heading angles are not exactly zero. Re-running this case using a better "bad guess,"

$$(t_f, \mu_f, \lambda_f, \gamma_f) = (600\,\text{sec}, 3^\circ, 0^\circ, 0^\circ) \tag{4.36}$$

verified that the difference in performance is negligible, a difference of only 0.025% in the optimal downrange solution (i.e., objective function value). Also note that when using a better guess, the bank and heading angles were exactly zero as expected in a nominal maximum downrange trajectory. Note that the initial geographical position of the vehicle is on the equator and the prime-meridian.

The following results were generated using a 20-node solution to bootstrap an 80-node run on a P4, 3.06 GHz, 1.05-GB RAM personal computer. The corresponding CPU runtimes for the 20-node and 80-node solutions were 29 sec and 52 sec, respectively.

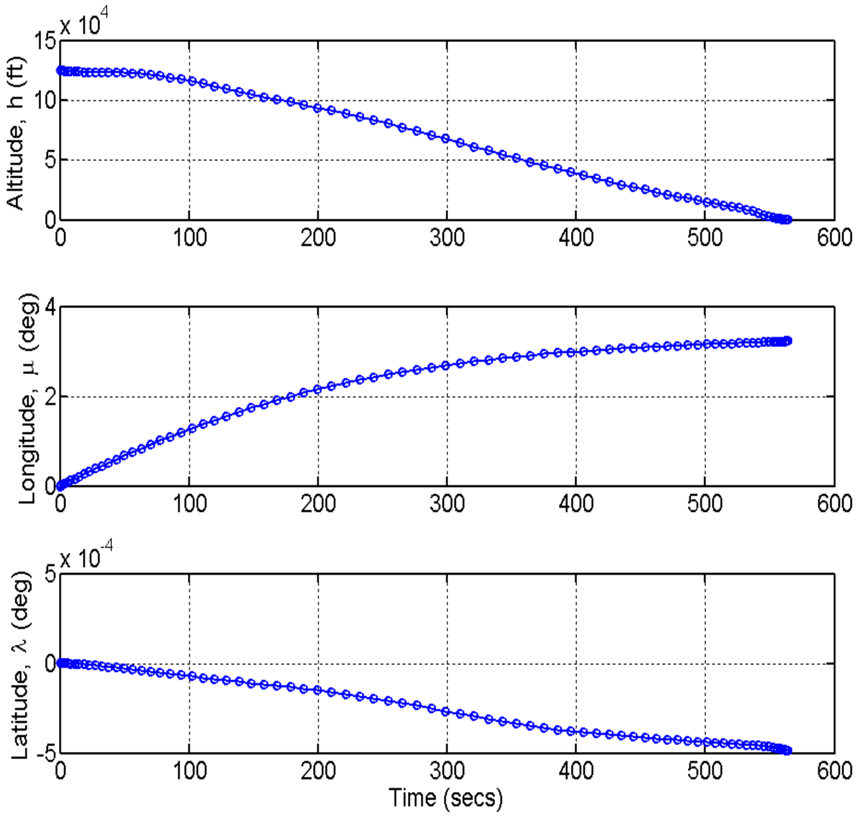


Figure 4.6 Open-Loop States for Max DR Solution.

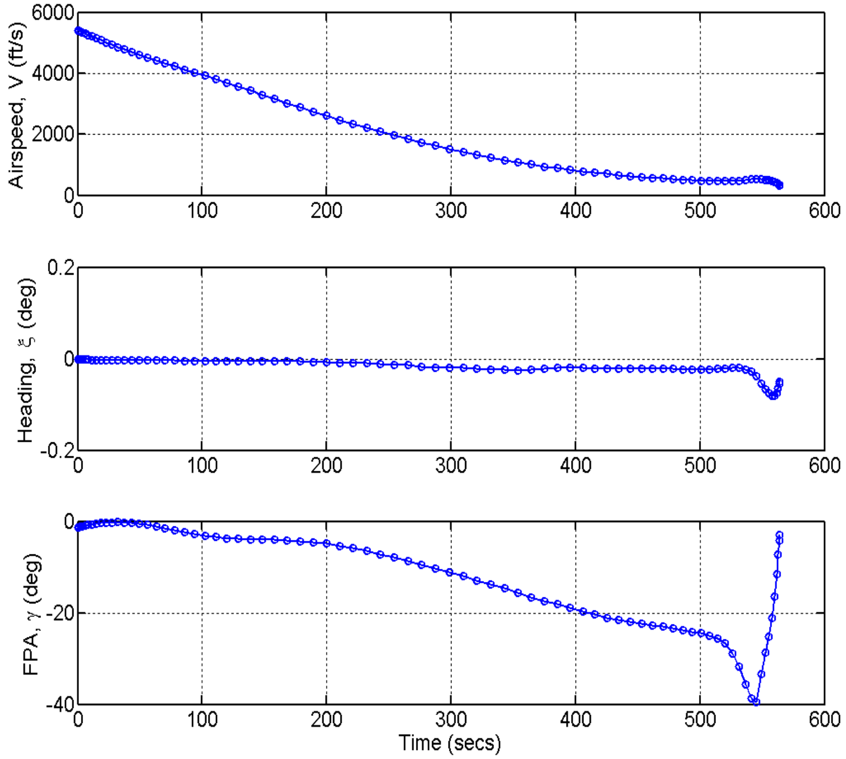


Figure 4.7 Open-Loop States for Max DR Solution.

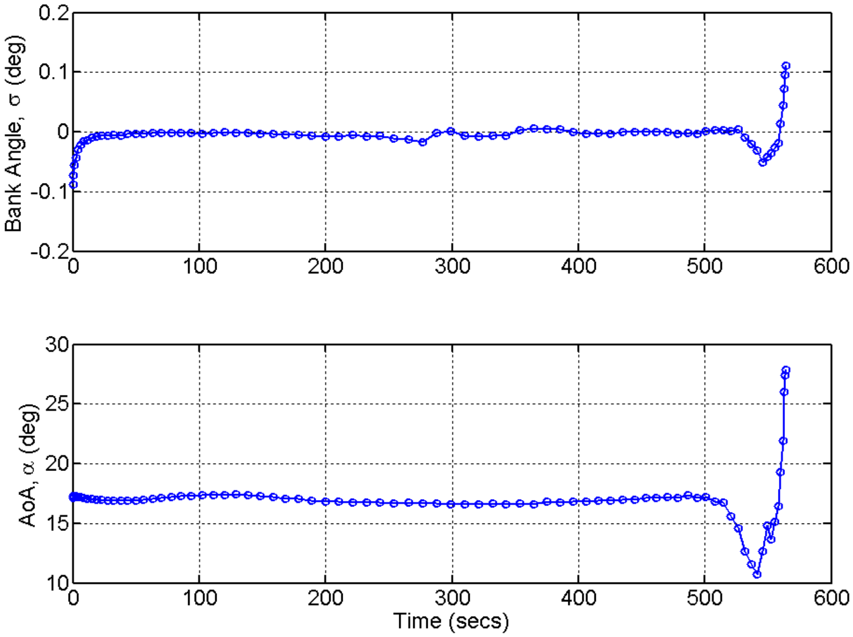


Figure 4.8 Open-Loop Controls for Max DR Solution.

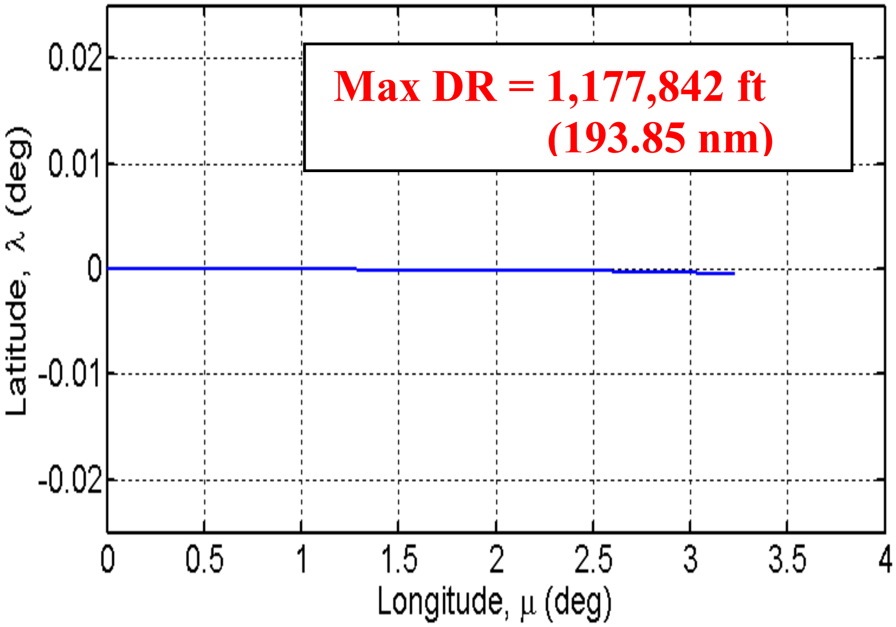


Figure 4.9 Open-Loop 2D Ground Track for Max DR Solution.

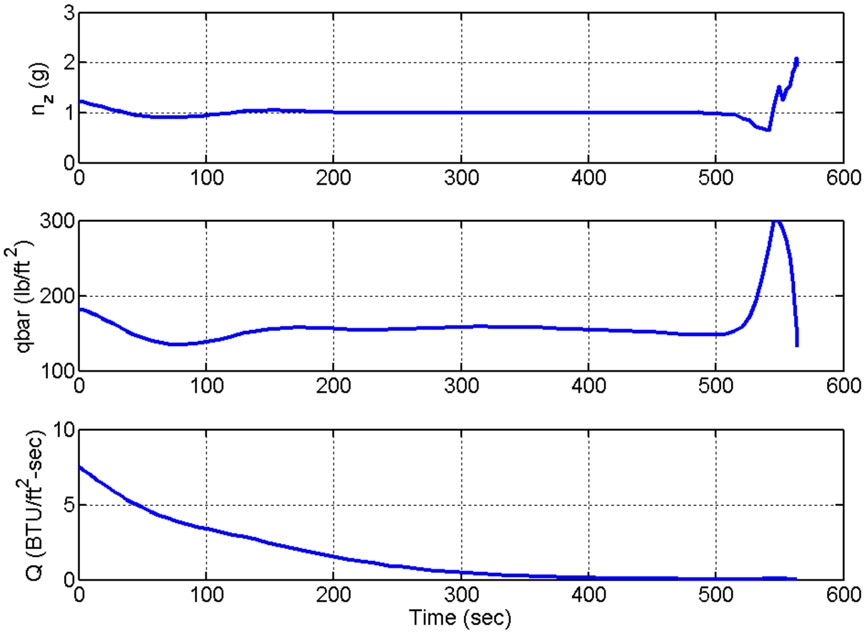


Figure 4.10 Open-Loop Path Functions for Max DR Solution

Table 4.2 20-Node vs. 80-Node Solution

	20 Nodes	**80 Nodes**	**Diff.**
RunTime (sec)	29.0152	52.0156	23.0004
Lon_f (deg)	3.2302	3.2308	0.0006
Lat_f (deg)	-0.0009	-0.0005	0.0004
DR (ft)	1177624.3230	1177842.0000	217.6770
DR (nm)	193.8120	193.8480	0.0360
CR (ft)	330.1302	182.2835	147.8467
CR (nm)	0.0543	0.0311	0.0232

As shown in Figure 4.9 and listed in Table 4.2, the maximum downrange is approximately 193.85 nm. Note that there is a small crossrange component of 0.0311 nm in this solution as a result of the extremely poor guess. As shown in Table 4.2, the effect of bootstrapping a low-node solution to a higher-node solution is slightly different answers. Although the computation time increases approximately 23 sec, there is noticeable improvement in the accuracy of the final solution as illustrated later when comparing the error norm between the DIDO and propagated solution. Figure 4.10 illustrates that the path constraints were indeed satisfied throughout the flight trajectory.

(2) Maximum Crossrange Performance. Next, the maximum crossrange is computed by minimizing the final crossrange distance as given by Eq. (4.16) and repeated here,

$$Min \;\; J(\cdot) \; = \lambda_f \tag{4.37}$$

For this problem, the same initial and final conditions are specified as the maximum downrange case,

$$\begin{aligned}
&(t_0, h_0, \mu_0, \lambda_0, V_0, \gamma_0, \xi_0) = (0\,\text{sec}, 125000\,\text{ft}, 0^\circ, 0^\circ, 5413\,\text{ft/s}, -1.3^\circ, 0^\circ) \\
&(h_f, V_f, \gamma_f) = (500\,\text{ft}, 335\,\text{ft/s}, -3.0^\circ)
\end{aligned} \tag{4.38}$$

Note again that the initial longitude and latitude uses the same poor guesses. Like the downrange case, the crossrange results were generated using a 20-node solution to bootstrap an 80-node run using the same computer. The corresponding CPU runtimes for the 20-node and 80-node solutions were 19 sec and 22 sec, respectively.

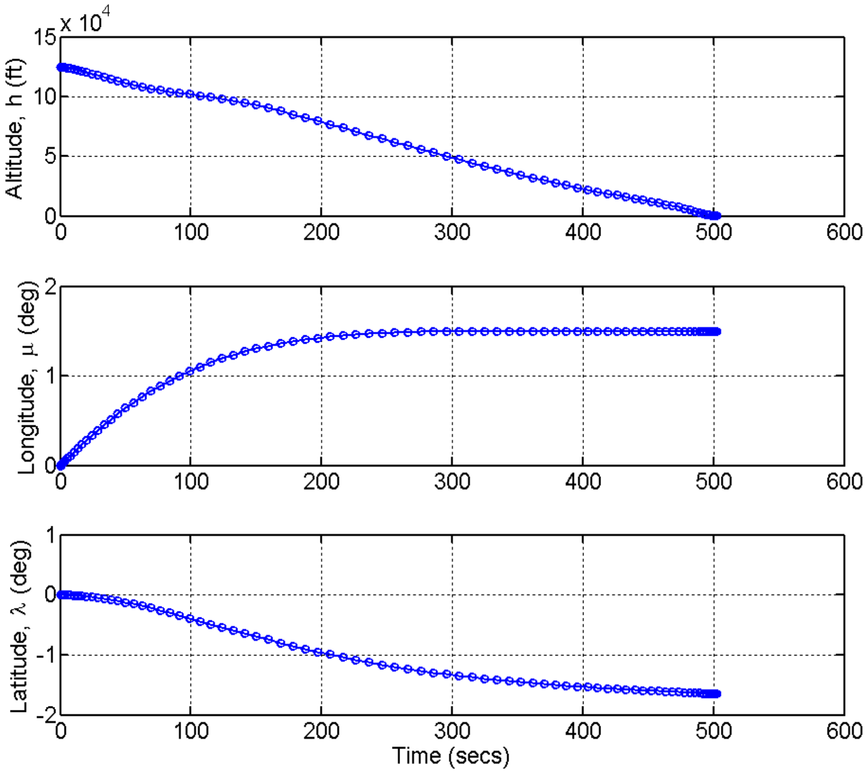


Figure 4.11 Open-Loop States for Max CR Solution.

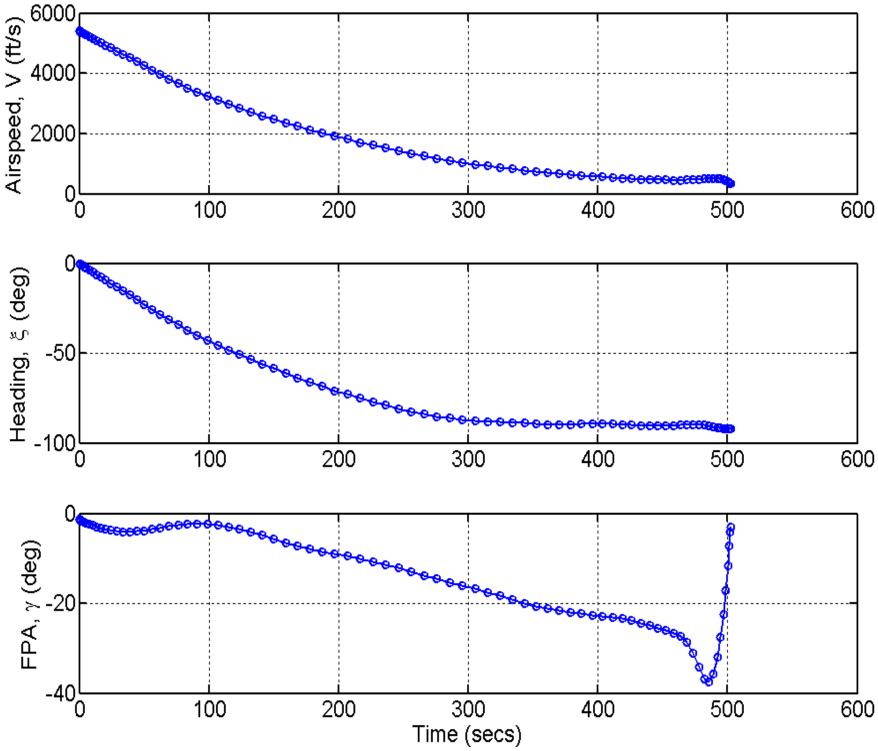


Figure 4.12 Open-Loop States for Max CR Solution.

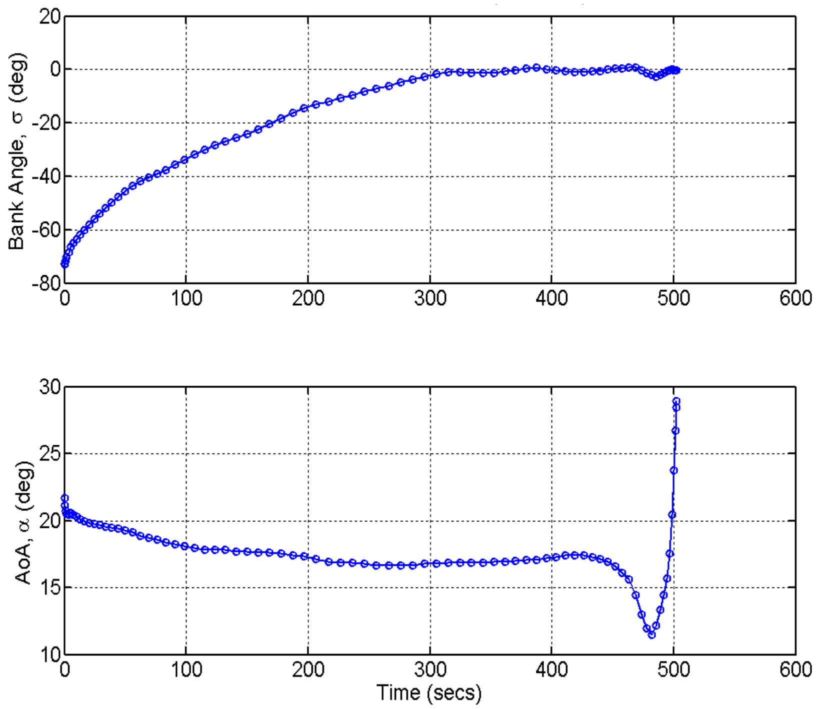


Figure 4.13 Open-Loop Controls for Max CR Solution.

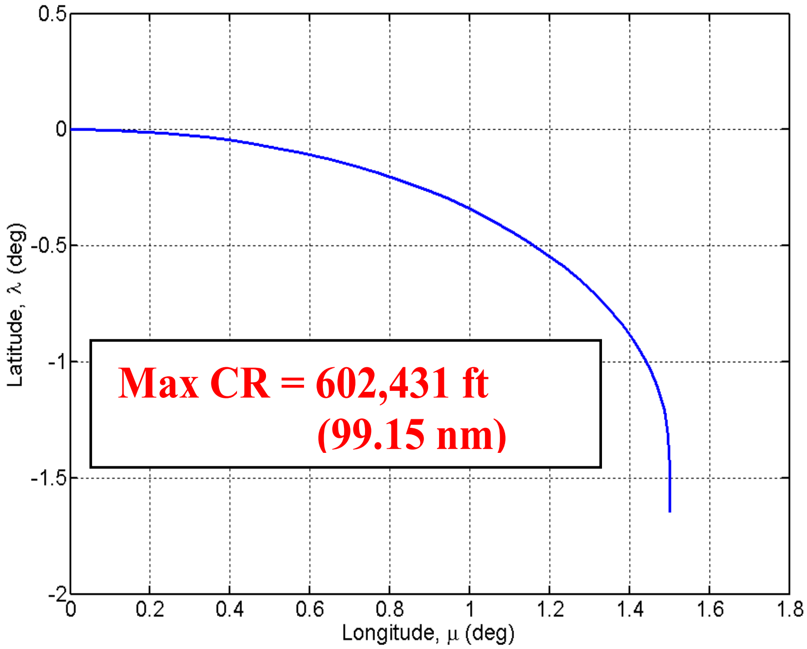


Figure 4.14 Open-Loop 2D Ground Track for Max CR Solution.

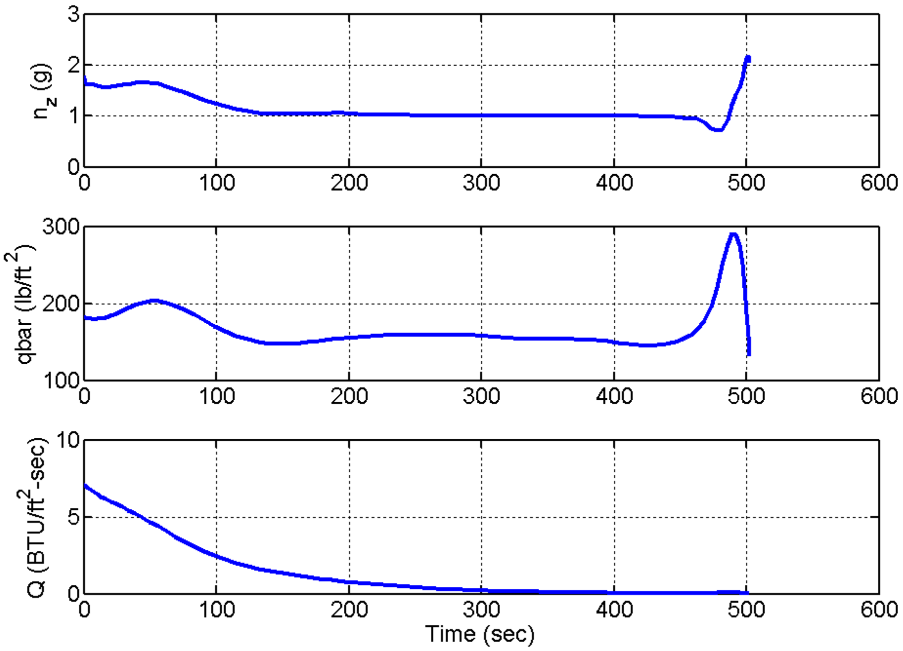


Figure 4.15 Open-Loop Path Functions for Max CR Solution.

Table 4.3 20-Node vs. 80-Node Solution

	20 Nodes	80 Nodes	Diff.
RunTime (sec)	22.8750	26.0156	3.1406
Lon_f (deg)	1.4998	1.5006	0.0008
Lat_f (deg)	-1.6515	-1.6525	0.0010
DR (ft)	546790.0000	547070.0000	280.0000
DR (nm)	89.9895	90.0362	0.0467
CR (ft)	602070.0000	602431.0000	370.0000
CR (nm)	99.0875	99.1515	0.0640

As shown in Figure 4.14 and listed in Table 4.3, the maximum crossrange is approximately 99.15 nm. As expected, the trajectory starts with the maximum bank angle until its heading reaches 90 deg from the equator and then flies straight, wings-level to maximize crossrange. Note that the bootstrapped solution only takes 3.14 sec longer than 20-node solution. Again, all the path constraints were satisfied. Note how for both cases, the maximum downrange and the maximum crossrange, the RLV automatically performs a flare-like “pull-up” maneuver towards the end of the trajectory as indicated by the FPA and AoA. This type of behavior illustrates how this method can be used for even the autoland guidance phase as previously described. Another interesting observation is that during this flare maneuver, the dynamic pressure nears its limit of 300 lb/ft^2 as shown in Figure 4.15. If a margin of safety is desired at this lower altitude, the dynamic pressure constraint can be tightened at the final condition using variable path constraint limits or an additional terminal event condition.

b. Feasibility and Optimality Analysis

The *feasibility* of the computational solution can be independently validated by comparing the DIDO results to the propagated states via a separate ODE Runge-Kutta propagator. By interpolating the values of the control function, $u(t_i)$, at the LGL points and then integrating the differential dynamical equations, $\dot{x} = f(x, u(t), t)$, via MATLAB's ode45 solver, a comparison of error norms can be made with the DIDO trajectory results. Figure 4.16 compares the DIDO and interpolated controls. As seen, there is very little difference. Depending on the structure of the curve, the use of different interpolating schemes (i.e., linear, cubic, spline, etc) may be more effective. For these control solutions, "cubic" interpolation works fine. Figure 4.17 and Figure 4.18 compare the DIDO solution with the ode45-propagated solution. As seen, there is very little difference between the two solutions; hence, confirming feasibility.

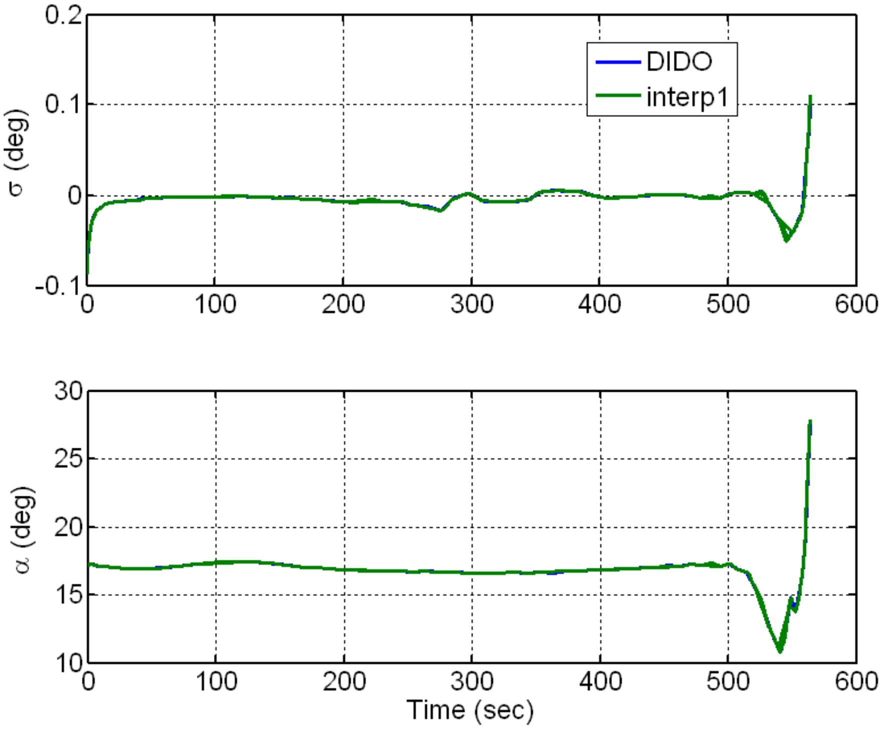


Figure 4.16 Interpolated Controls for Max DR Solution.

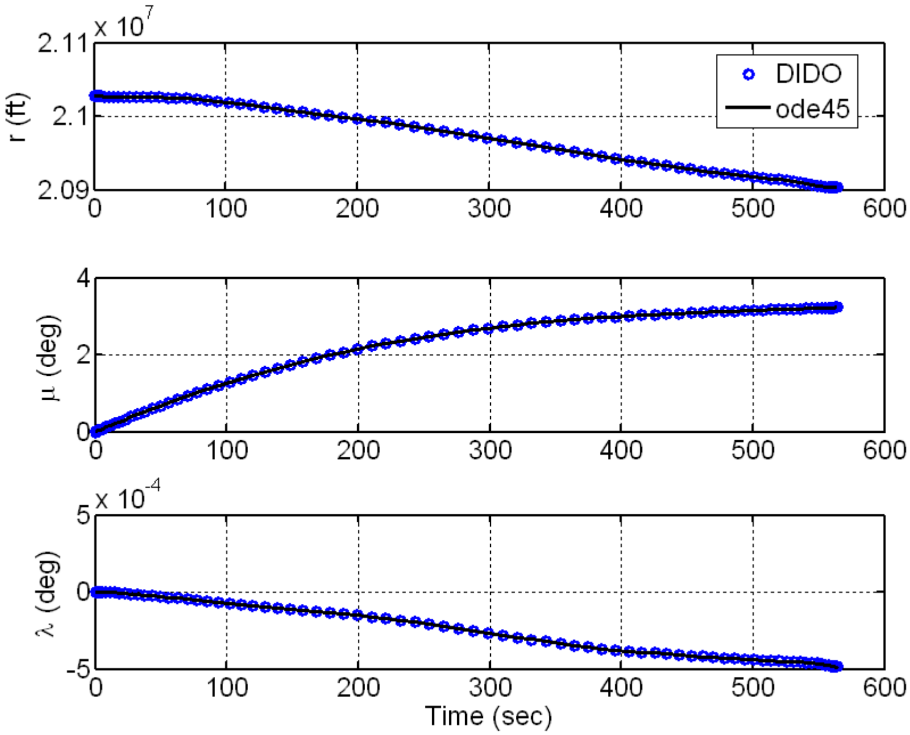


Figure 4.17 Comparison of DIDO and Propagated States for Max DR Solution.

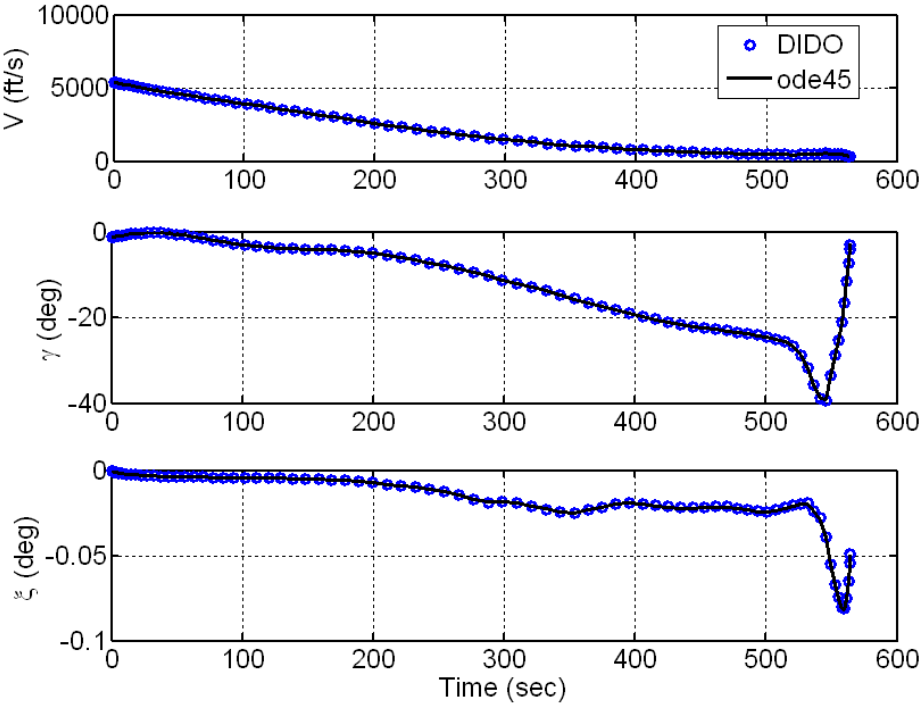


Figure 4.18 Comparison of DIDO and Propagated States for Max DR Solution.

Table 4.4 shows the norm error when comparing the DIDO solution, $\mathbf{x}_{\text{DIDO}}$, to the propagated solution, $\mathbf{x}_{\text{Prop}}$ such that $\Delta\mathbf{x}_f = \mathbf{x}_{\text{Prop}} - \mathbf{x}_{\text{DIDO}}$. Notice that the difference for the altitude is the largest with an absolute error of 328.97. By increasing the number of nodes from 20 to 80, the accuracy of the final altitude improves from 331.16 ft to 2.19 ft, respectively.

Table 4.4 Final State Vector Errors for 20-Node and 80-Node Solution

	20 Nodes	**80 Nodes**	**Diff.**
Δh_f (ft)	331.1585	2.1866	328.9719
$\Delta\mu_f$ (deg)	0.0008	-0.0001	0.0009
$\Delta\lambda_f$ (deg)	0.0000	0.0000	0.0000
ΔV_f (ft/s)	-2.0210	-0.9146	1.1064
$\Delta\gamma_f$ (deg)	-0.9939	0.2984	1.2923
$\Delta\xi_f$ (deg)	0.0048	-0.0001	0.0049

One way to demonstrate optimality is by verifying that the necessary optimality conditions are satisfied. This is performed by analytically solving some of the necessary conditions and then comparing the analytic results with the numerical solutions. In the absence of useful costate information, *Bellman's Principal of Optimality* can be applied to validate optimality. Both of these optimality tests are demonstrated next.

Comparing the numerical results to the theoretical analysis of the HVC, HEE, HMC, and TTC conditions validates the *optimality* of the computational solution. The HVC stated in the theoretical analysis indicates that the Hamiltonian should be zero at the final time (i.e., $H(t_f) = 0$). From the HEE, it can be shown that the Hamiltonian is constant with respect to time. Combining these two conditions, the Hamiltonian should be zero for all time, clearly evident in Figure 4.19.

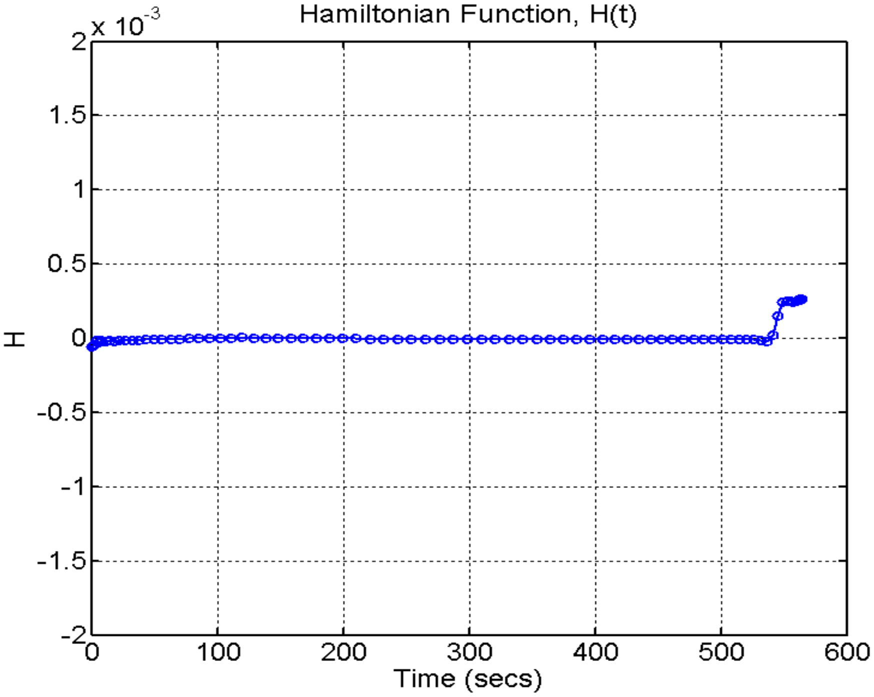


Figure 4.19   Hamiltonian for Max DR Solution.

Another test to confirm computational optimality is to apply *Bellman's Principle of Optimality*. This principle essentially states that by using any point on the original optimal trajectory as an initial condition to a new problem, with all other problem formulation parameters the same, should result in the same optimal trajectory with the same or better cost. This method was used to validate the open-loop optimality as demonstrated in Figure 4.20.

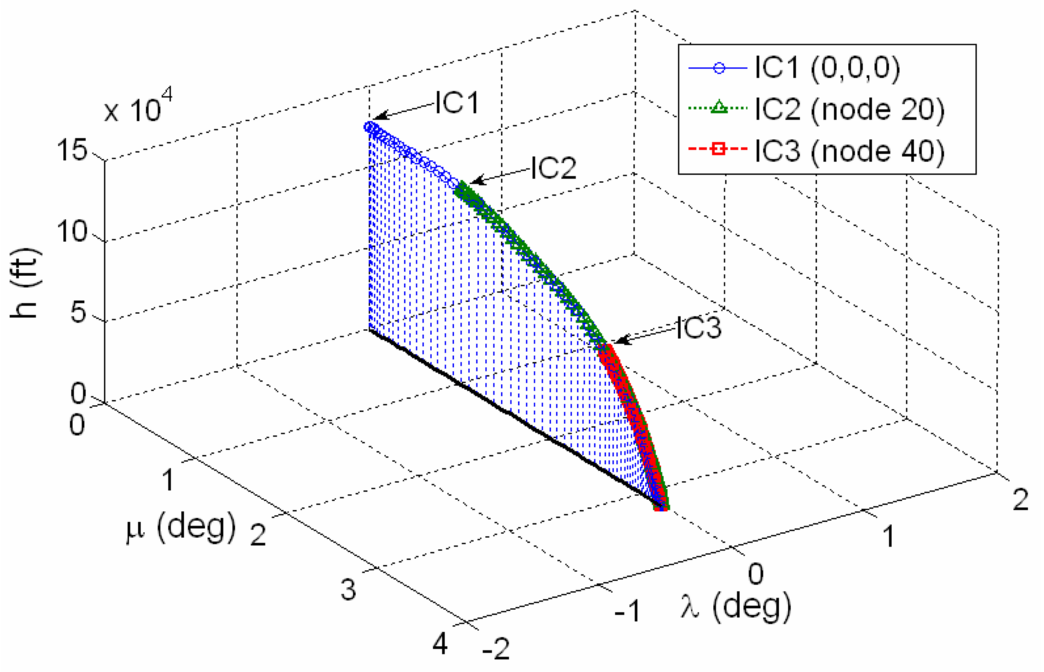


Figure 4.20 Bellman Test for Max DR Solution.

Similarly, the following figures are used to numerically confirm feasibility and optimality of the maximum crossrange solutions.

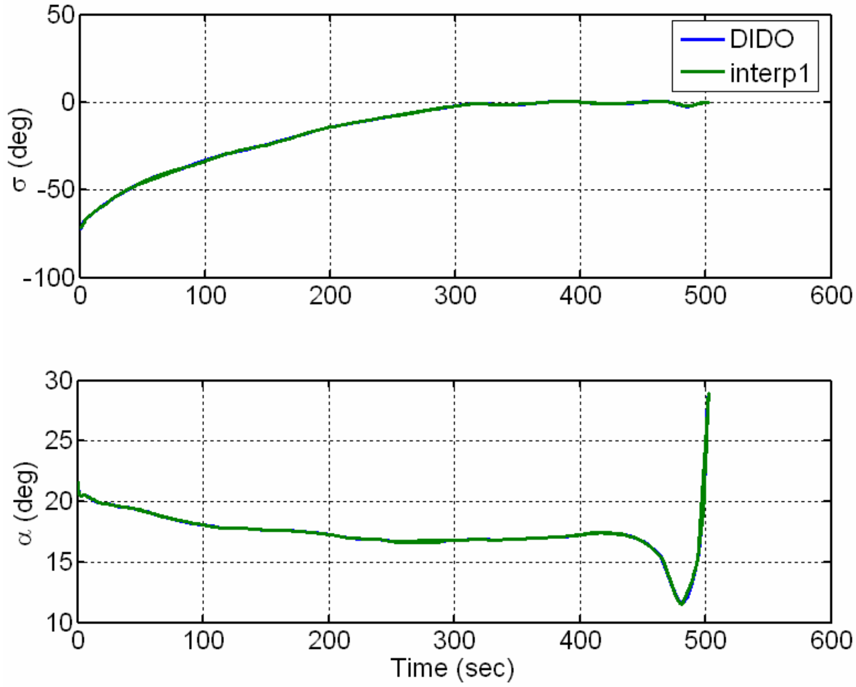


Figure 4.21 Interpolated Controls for Max CR Solution.

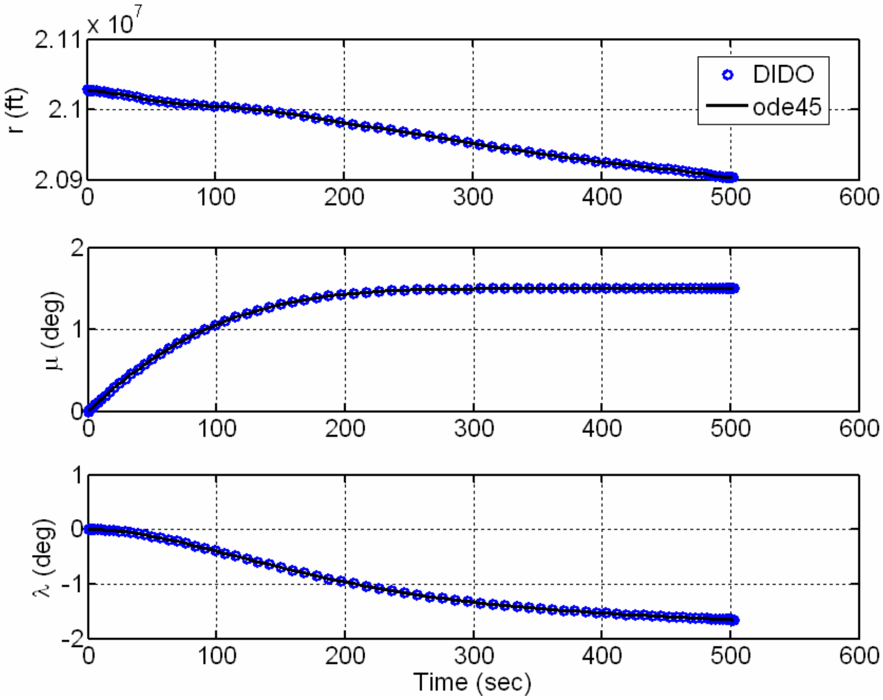


Figure 4.22 Comparison of DIDO and Propagated States for Max CR Solution.

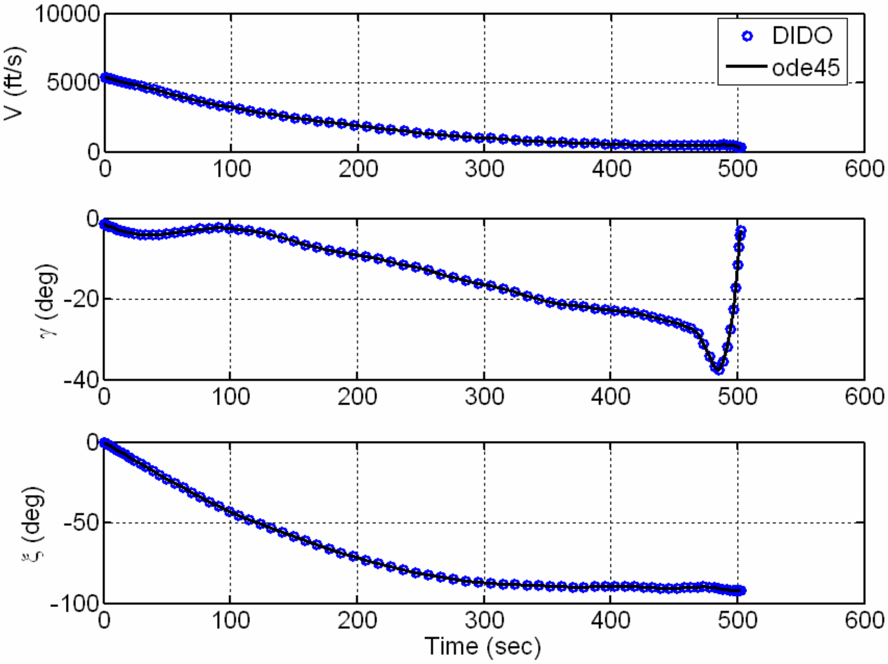


Figure 4.23 Comparison of DIDO and Propagated States for Max CR Solution.

Table 4.5 shows the norm error when comparing the DIDO solution to the propagated solution for the maximum crossrange case. Notice that the difference for the altitude is again the largest with an absolute error of 231.21. By increasing the number of nodes from 20 to 80, the accuracy of the final altitude improves from approximately 230.27 ft to 0.95 ft, respectively.

Table 4.5 Final State Vector Errors for 20-Node and 80-Node Solution

	20 Nodes	**80 Nodes**	**Diff.**
Δh_f (ft)	230.2667	-0.9468	231.2135
$\Delta \mu_f$ (deg)	-0.0015	-0.0003	0.0012
$\Delta \lambda_f$ (deg)	-0.0007	0.0001	0.0008
ΔV_f (ft/s)	-3.6967	-0.1353	3.5614
$\Delta \gamma_f$ (deg)	-0.4948	-0.0078	0.4870
$\Delta \xi_f$ (deg)	-0.3107	-0.0257	0.2850

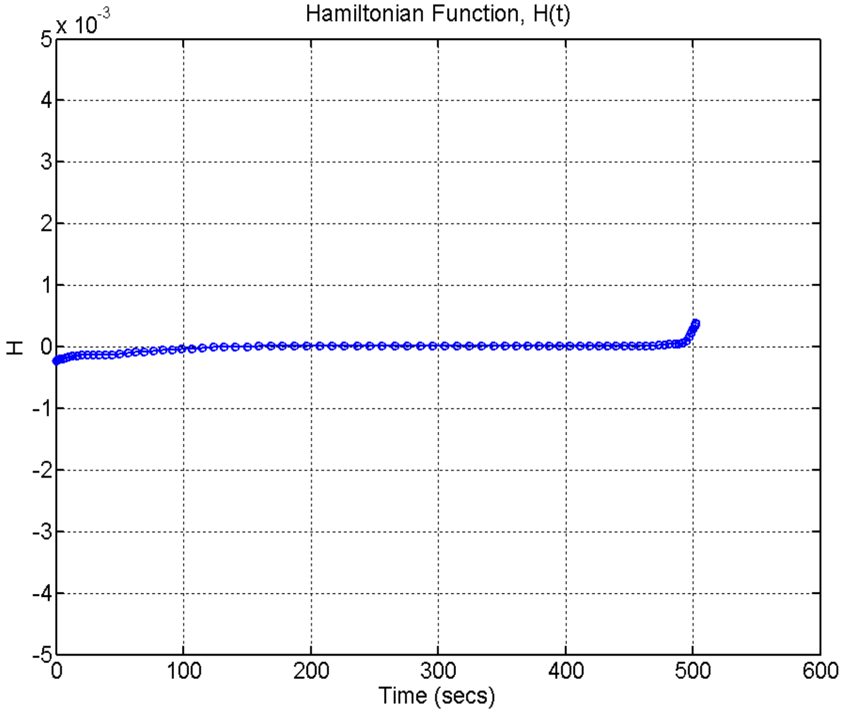


Figure 4.24 Hamiltonian for Max CR Solution.

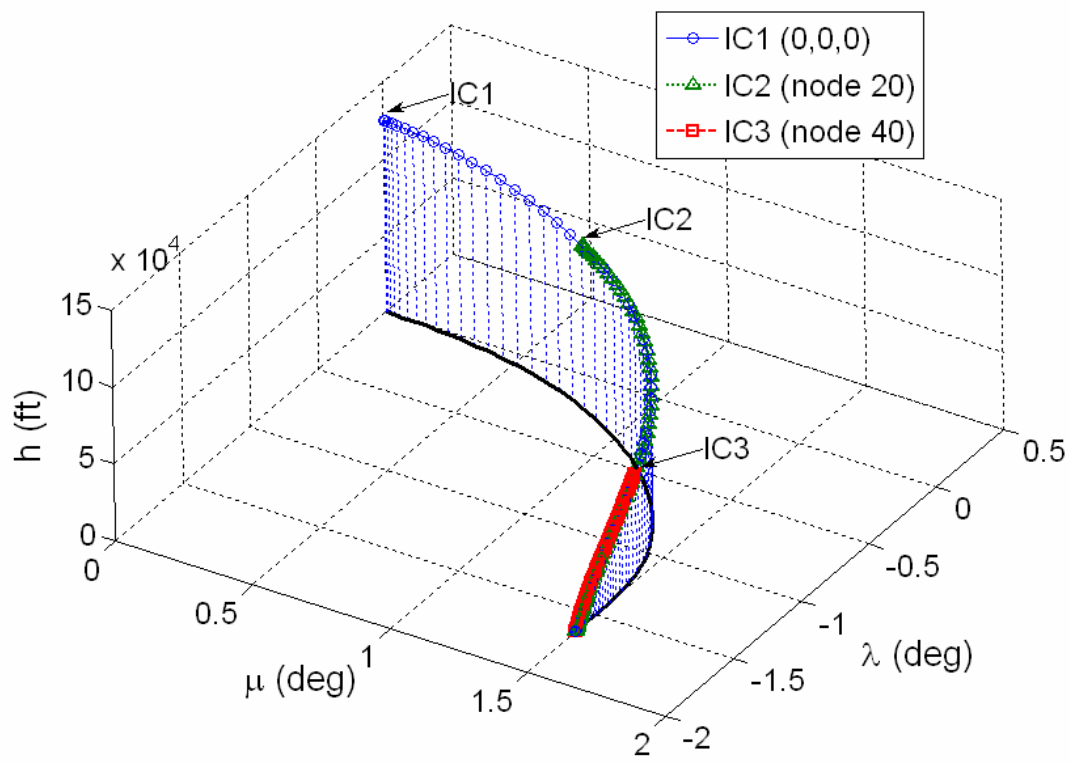


Figure 4.25 Bellman Test for Max CR Solution.

c. *Robust and Intelligent Trajectory Generation*

This section gives an example of intelligent reentry guidance for an X-33-class RLV. The objective of the problem is to optimally plan a path from the reentry point (suborbital in this case) to the landing site or some designated end-point in the vicinity thereof. As previously stated, the complexity of this mission is driven by the drastically changing environment that the vehicle undergoes while transcending the atmosphere. Of primary concern are the heating rate, the dynamic pressure, and the structural loads placed on the vehicle as a result of hypersonic speeds through large altitude-density variations. The following example demonstrates the proposed method's ability to rapidly generate feasible and optimal trajectories to various landing sites in Florida. Historically, as in the case of Space Shuttle guidance, it takes many man-hours of pre-flight planning to design such trajectories off-line and then pre-program the Shuttle's flight computer with numerous waypoints and contingency trajectories. Although one of the most astonishing designs in aerospace history, the Space Shuttle lacks the autonomous capability to generate new trajectories "on-the-fly." This limitation creates a potentially lethal safety risk and as such, has fueled tremendous efforts to improve the guidance and capabilities of future RLVs. This example demonstrates how optimal trajectory generation injects elements of intelligence into now, autonomous path planning.

Posing the final target as a reachable set defined by certain state-control constraints, eliminates the need for detailed trajectory design. By simply stating this "reachable set" as a goal, the guidance system should be able to determine how to accomplish the goal and then activate appropriate steps needed to achieve that goal (i.e., control). This is a form of intelligent behavior since it consists of autonomous planning and execution.

First, to verify that the trajectory generation method is robust in terms of convergence of feasible trajectories, the initial conditions are varied while keeping all other parameters the same. The FAC-target in this case is specified as the center of a 2D-box generated from a planar-projection of the FAC as defined in Figure 4.26. With this, all the final endpoint conditions corresponding to a landing approach at the Shuttle Landing Facility, Kennedy Space Center (KSC), Florida, are strict equalities given by

$$(h_f, \mu_f, \lambda_f, V_f, \gamma_f, \xi_f) = (2000\,\text{ft}, -80.7112^\circ, 28.6439^\circ, 300\,\text{ft/s}, -6.0^\circ, -60^\circ) \quad (4.39)$$

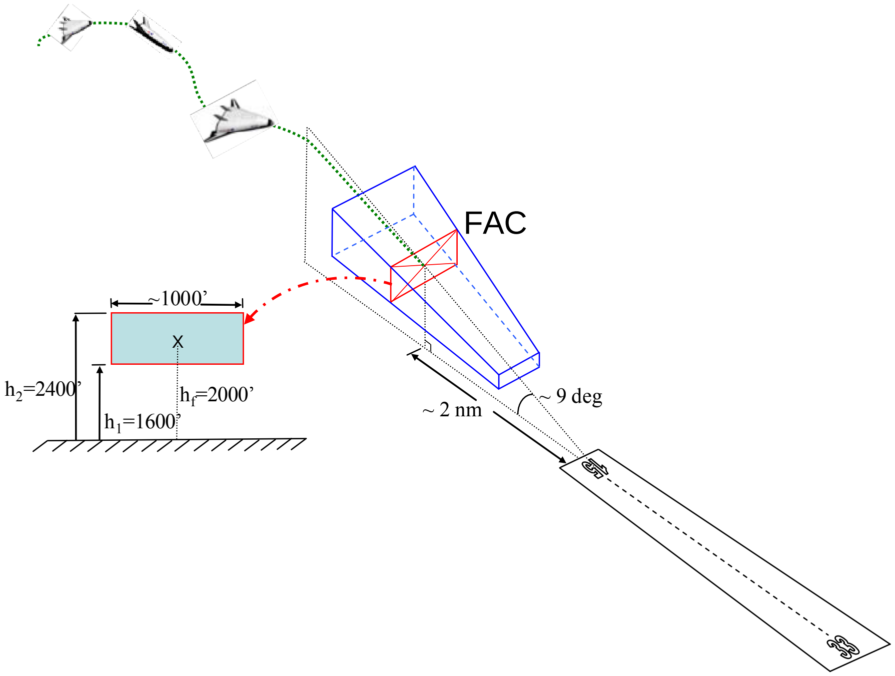


Figure 4.26 2D-Projection of Final Approach Corridor (FAC) for Strict "Target" Conditions.

The selection of initial conditions is somewhat arbitrary, but some are based on conditions that approximate the individual trajectory segments used by the Shuttle guidance system as discussed in Chapter I. As a reference, the communication black-out ends at approximately 162,000 ft, the TAEM interface occurs at about 83,000 ft, and the approach and landing phase initiates at about 10,000 ft. Table 4.6 lists the various initial conditions used for this analysis.

Table 4.6 Initial Conditions for Trajectories to FAC

IC	h_0 **(ft)**	μ_0 **(deg)**	λ_0 **(deg)**	V_0 **(ft/s)**	γ_0 **(deg)**	ξ_0 **(deg)**	α_0 **(deg)**	σ_0 **(deg)**
1	167323	-85	30	8530	-1.5	0	30	0
2	167323	-84	26	8530	-1.5	0	30	0
3	125000	-82	30	5413	-1.3	0	19	0
4	125000	-81.5	28.4	5413	-1.3	0	19	0
5	125000	-81.1	28.6	5413	-1.3	0	19	0
6	85000	-81.7	29	2800	-10	0	10	0
7	85000	-81.2	29	2800	-10	0	10	0
8	85000	-81.8	28.64	2800	-10	0	10	0
9	10000	-82	30	350	-5	0	5	0
10	10000	-80.73	28.75	350	-5	-40	5	0

Table 4.7 provides the numerical performance for the various initial conditions. The number of iterations and the corresponding CPU times are listed for the initial start-up nodes (20) and the bootstrapped nodes (40).

Table 4.7 Performance for Various Initial Conditions

	Cold 20 Nodes		Bootstrapped 40 Nodes	
IC	**Iterations**	**Time (sec)**	**Iterations**	**Time (sec)**
1	1326	11.2	207	6.7
2	1408	23	216	7.2
3	5767	49	938	15
4	2814	14.5	242	5.6
5	8571	32.1	4909	104
6	1269	11.2	206	5.6
7	5119	15.6	4569	22.5
8	3034	18.2	236	5.6
9	5822	29	592	30
10	6232	32	613	33

The average number of iterations and CPU times are approximately 2568 and 23.6 sec, respectively. The open-loop trajectories for the various initial conditions are shown in Figure 4.27 and Figure 4.28. As seen, the vehicle successfully intercepts the center of the FAC. Taking a closer look at the 4th trajectory, as shown in Figure 4.29, reveals a path that resembles typical aircraft-like operations. This exemplifies an intelligent system – emulating human expert behaviors in the sense of duplicating what human pilots have come to consider the standard through many years of experience. Since the velocity for an RLV is much larger than an aircraft, it makes sense that more drastic energy management is required. For this particular trajectory, the initial velocity and altitude are 5413 ft/s (M=5.33) and 125,000 ft, respectively. According to this path, as shown by the 2D ground track in Figure 4.30, when the vehicle is within 4.23 nm of the runway, it is still traveling at 1924 ft/s (M=1.94). In order to bleed-off this velocity, the algorithm determines that a bank-reversal is required that extends the flight away from the runway, then another to turn back towards the runway with just enough velocity to satisfy the final conditions.

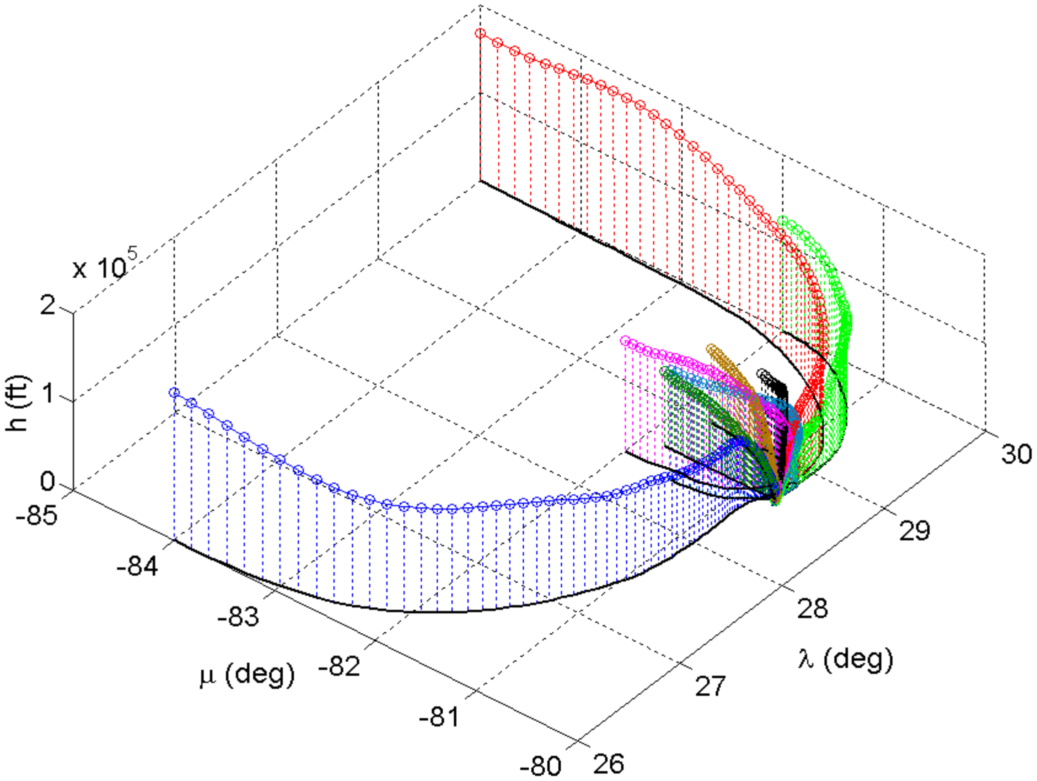


Figure 4.27 Robustness to Variations in Initial Conditions.

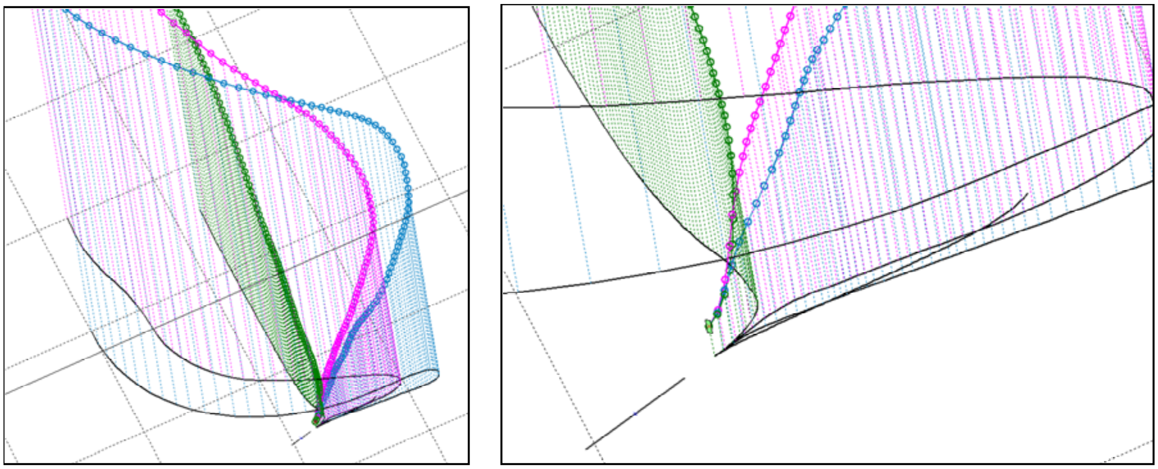


Figure 4.28 Alternate View to Show Convergence to FAC.

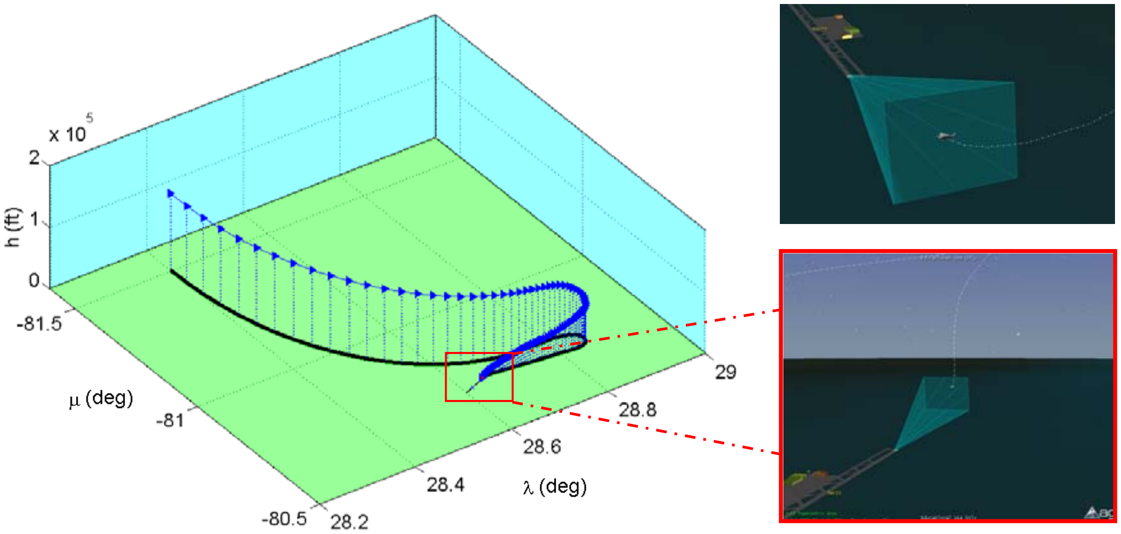


Figure 4.29   Open-Loop Trajectory Generated Autonomously to Intercept FAC.

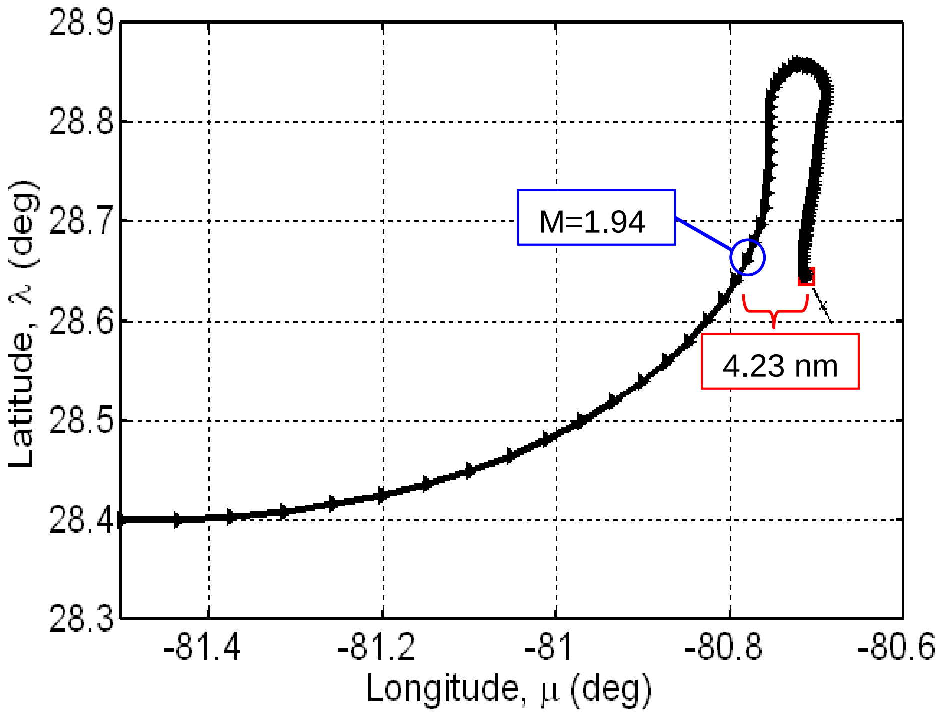


Figure 4.30   2D Ground Track of Trajectory # 4.

The next set of plots, Figure 4.31 and Figure 4.32, are results from simulating an in-bound vehicle approaching the designated landing site from multiple latitudes (27.5 to 30.0 deg), which may be the case for a vehicle that has de-orbited from different inclinations. For this case, the initial longitude is fixed over the Gulf-of-Mexico at -82 deg and the final conditions are the same as given in Eq. (4.39). Like the previous results, all the trajectories converge to the center of the FAC, appropriately aligning the vehicle with the runway (heading of 150 deg).

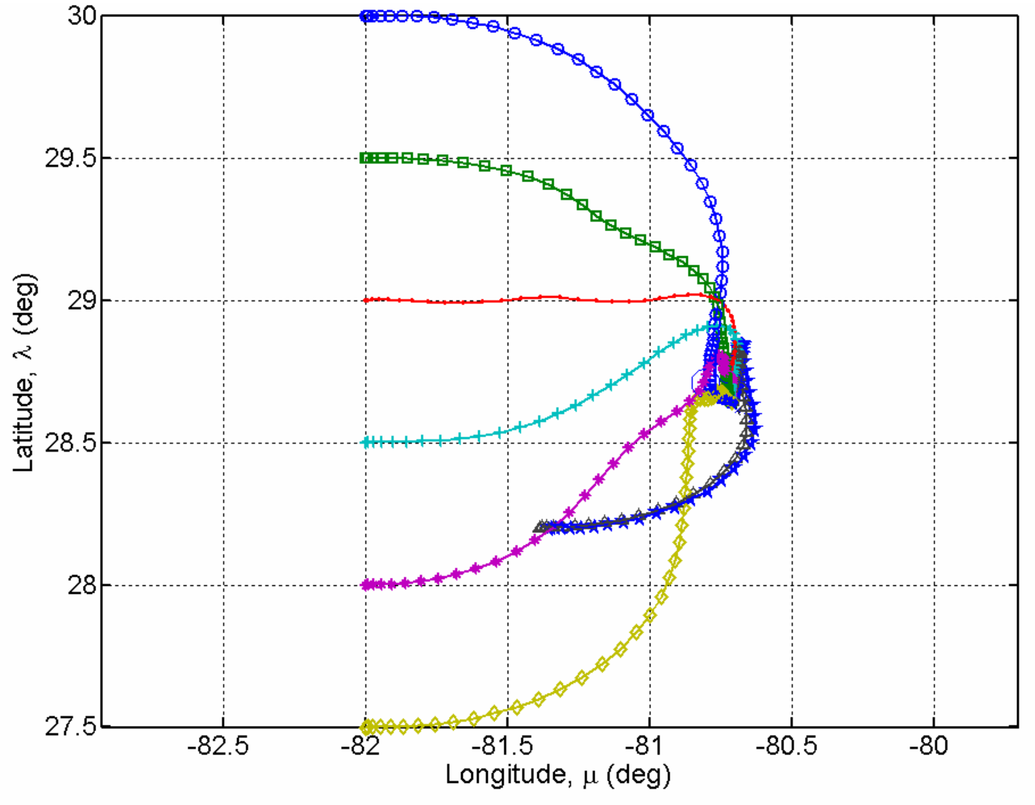


Figure 4.31 Robustness to Variations in Initial Conditions.

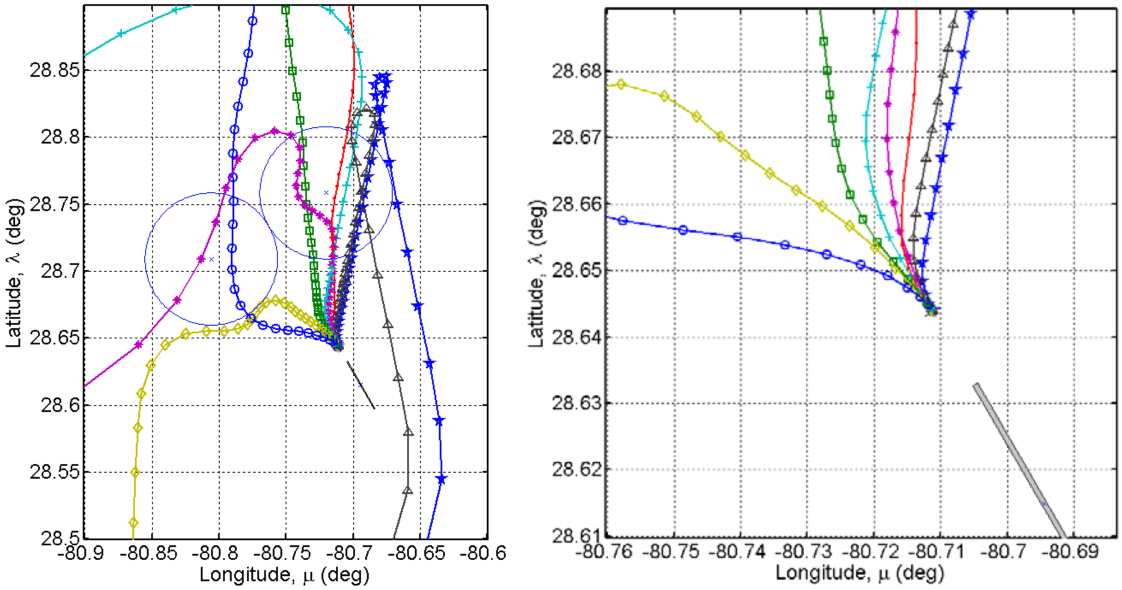


Figure 4.32 Alternate Zoomed View to Show Convergence to FAC.

These two examples present a family of feasible trajectories via pseudo-Monte-Carlo analysis that demonstrates the reliability of the optimization method without relying on TAEM or HAC (Heading Alignment Cylinder) trajectory segmentation. Now, the autonomy/intelligence of the method can be demonstrated by analyzing yet another 3-DOF example.

For the proceeding example, the optimal trajectory control sequence for a series of flight paths is again generated using the 3-DOF model given by Eq. (4.1). The specific initial and final conditions are given by

$$\begin{gathered}(t_0, h_0, \mu_0, \lambda_0, V_0, \gamma_0, \xi_0) = (0\,\text{sec}, 85000\,\text{ft}, -81^\circ, 28.7^\circ \text{ and } 28.6^\circ, \mathbf{V}^0, -10^\circ, 5^\circ) \\ (t_f, h_f, \mu_f, \lambda_f, V_f, \gamma_f, \xi_f) = (1000\,\text{sec}, 2000\,\text{ft}, -80.7112^\circ, 28.6439^\circ, 300\,\text{ft/s}, -6.0^\circ, -60^\circ)\end{gathered} \tag{4.40}$$

where the initial velocity, indicated by $\mathbf{V}^0$, is increased from 400 ft/s to 2900 ft/s. Both scenarios initiate approximately 60 nm from the runway with the first slightly north-west $(28.7^\circ, -81.0^\circ)$ and the second slightly south-west $(28.6^\circ, -81.0^\circ)$. Note that the initial altitude coincides with the TAEM interface used in Shuttle guidance. The results, presented in Figure 4.33 and Figure 4.34, illustrate a good example of intelligent trajectory generation. As shown by the first trajectory (dark green) with an initial velocity of 400 ft/s, the trajectory emulates a typical "direct, straight-in" approach towards the runway. With limited energy, the vehicle guidance system (i.e., optimal path planning method), intelligently chooses a path that flies directly to the runway in order to satisfy the required endpoint conditions. As the initial energy increases, the trajectories display more maneuvers as indicated by the path with an initial velocity of 1600 ft/s. In this case, the vehicle has more energy than needed and generates a trajectory that contains "S-turn"-like maneuvers. As the energy becomes very large, the planned path actually requires the vehicle to turn away from the runway, or over-fly it, and then loop back around as indicated by the last two trajectories. This is similar to the shuttle's "overhead approach" procedure that is pre-programmed into the guidance system. Here, the guidance system did not need to be told what to do.

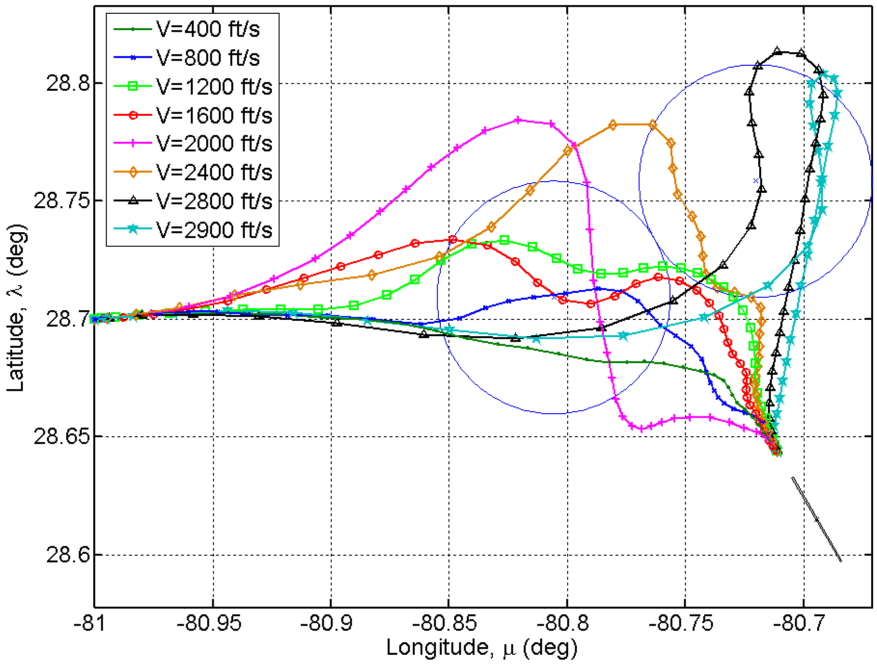


Figure 4.33 Intelligent Path Planning - Autonomously Accounts for Energy by Transitioning from "Direct, Straight-In" to "S-Turn" to "Overhead" Approaches ($\lambda_0 = 28.7^{\circ}$).

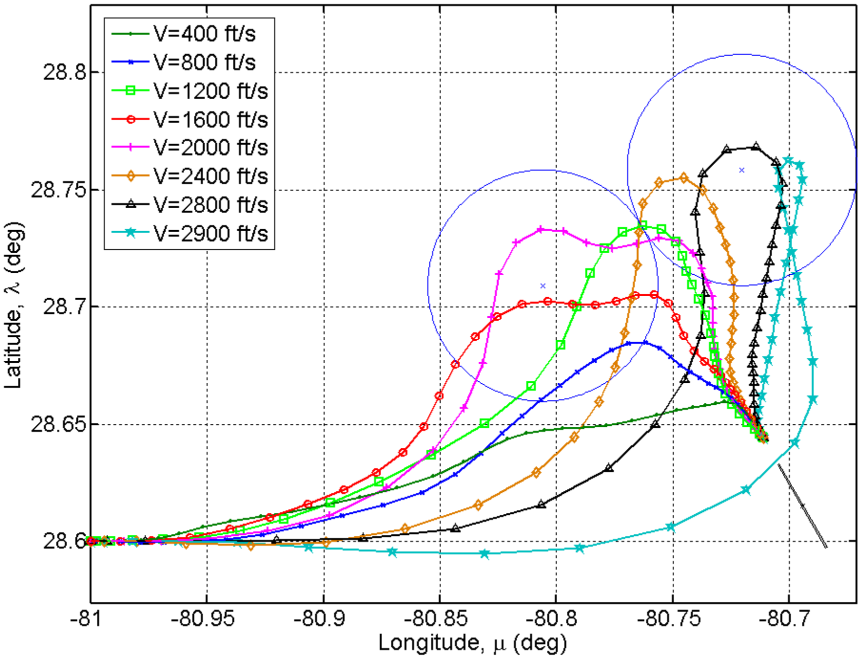


Figure 4.34 Intelligent Path Planning - Autonomously Accounts for Energy by Transitioning from "Direct, Straight-In" to "S-Turn" to "Overhead" Approaches ($\lambda_0 = 28.6^\circ$).

The blue circles in Figure 4.34, represent the HACs used in Space Shuttle guidance [16]. The purpose of overlaying the HACs on the plot is to see if there are any similarities between the optimal trajectories generated using the PS method and the typical trajectories that the Space Shuttle would track based on the HAC waypoints. For example, using a similar energy profile as the Space Shuttle, Figure 4.35 shows that the optimal trajectory generation actually seems to use the HAC, but the algorithm has no knowledge of such HAC. This clearly demonstrates the power of using optimal control for autonomous and intelligent applications.

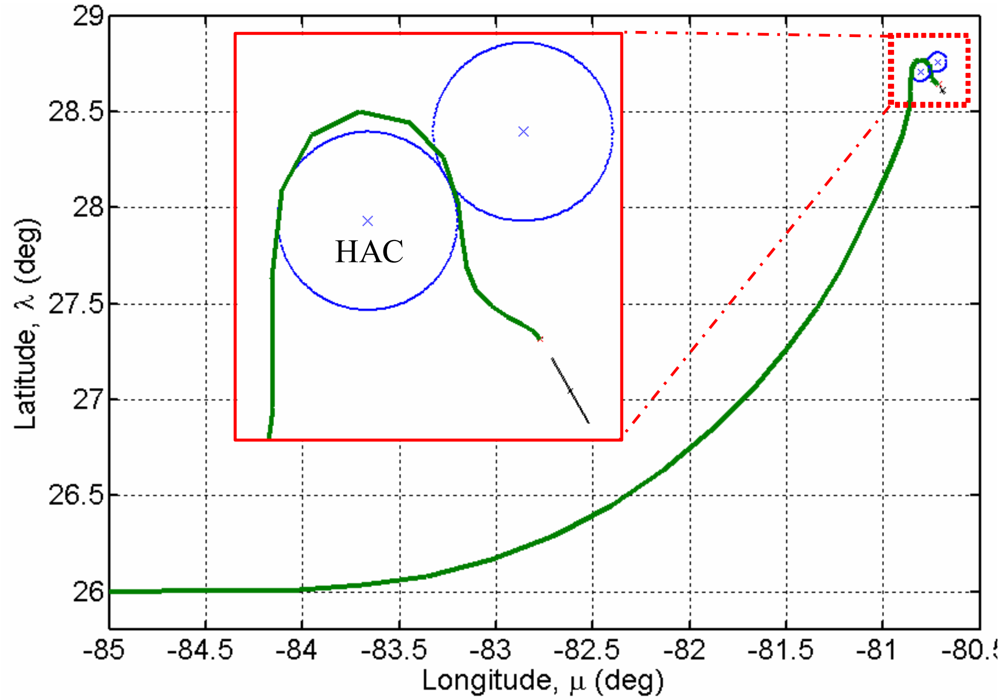


Figure 4.35 Example of Trajectory that Automatically Duplicates Effect of Space Shuttle HAC.

This example clearly shows that pre-determined waypoints or reference trajectories are not necessarily needed, but if they were desired, this method can be used to generate such reference profiles needed for tracking algorithms.

d. Footprint Generation

In a situation that requires aborting from a planned trajectory due to vehicle failures, unpredicted uncertainties, or large disturbances, the guidance system must be capable of re-targeting another landing site, especially if reconfigurable control methods cannot provide adequate compensation. As a part of the re-targeting algorithm, the guidance system must be capable of generating landing footprints in order to select a feasible landing site. Obviously, to be of practical use, this footprint generation must be computed extremely fast. In addition, the computation of the footprints assumes the vehicle can accurately identify and model the effects of the damage (via a health monitoring and failure identification sub-system). Figure 4.36 shows the individual nominal 2D ground tracks that form the landing footprint for the 3-DOF model. Note that the footprint is actually a polygonal approximation to the true footprint.

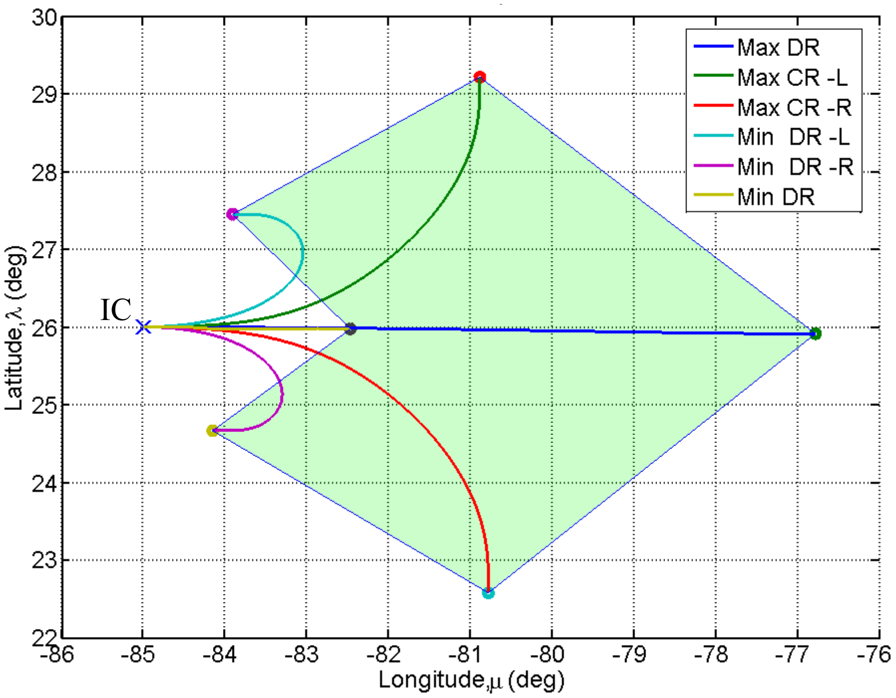


Figure 4.36 3-DOF Polygonal Footprint using Aerodynamic Approximation.

The aerodynamic models for these runs use the approximations given by the polynomial curve-fits of Eqs. (4.4) and (4.5). As shown, the trajectories for the left-turn minimum downrange (Min Dr –L) and the right-turn minimum downrange (Min Dr –R) indicate that the vehicle will actually perform 180-degree turns, reversing direction, in order to decrease its minimum range.

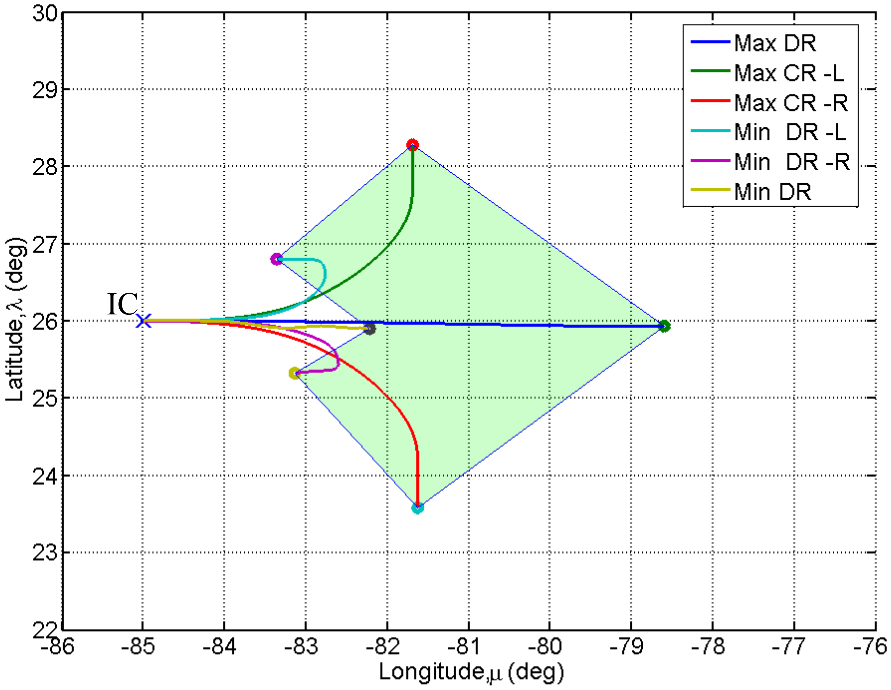


Figure 4.37 3-DOF Polygonal Footprint using Aerodynamic Tables.

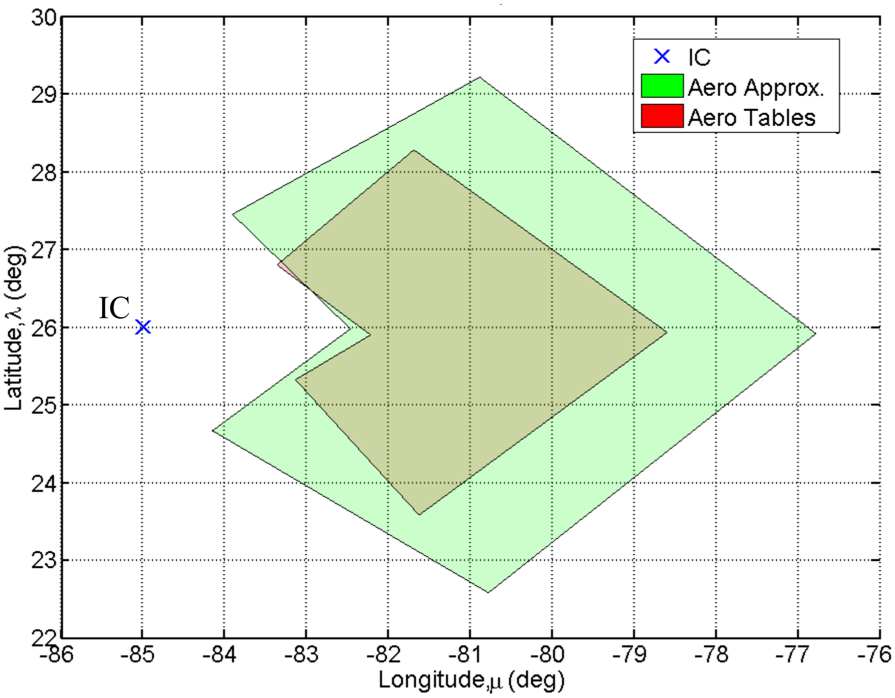


Figure 4.38 Comparison of Nominal 3-DOF Polygonal Footprints.

Note that the accuracy of the footprint is dependent on how many trajectories are generated. Ideally, the generation of these range-performance footprints would consist of formulating the cost functional as a weighted combination of maximum downrange and maximum crossrange (or minimum), a pareto-optimization problem, given by

$$J(\cdot) = w(\pm\mu_f) + (1-w)(\pm\lambda_f) \tag{4.41}$$

where μ_f and λ_f are the final downrange (longitude) and crossrange (latitude), respectively, and w is a weighting term such that $w \in [0,1]$. The accuracy of the footprints depends on the step size used to vary the weighting terms from 0 to 1 such that the approximate footprint approaches the exact footprint in the limit as $n \to \infty$, where n is the number of iterations. For example, repetitively solving the optimization problem while incrementally varying w will provide a footprint depicted in Figure 4.39 by the dotted line. Note that the difference between the linear footprint segments and the curved segments can be on the order of 60 nm.

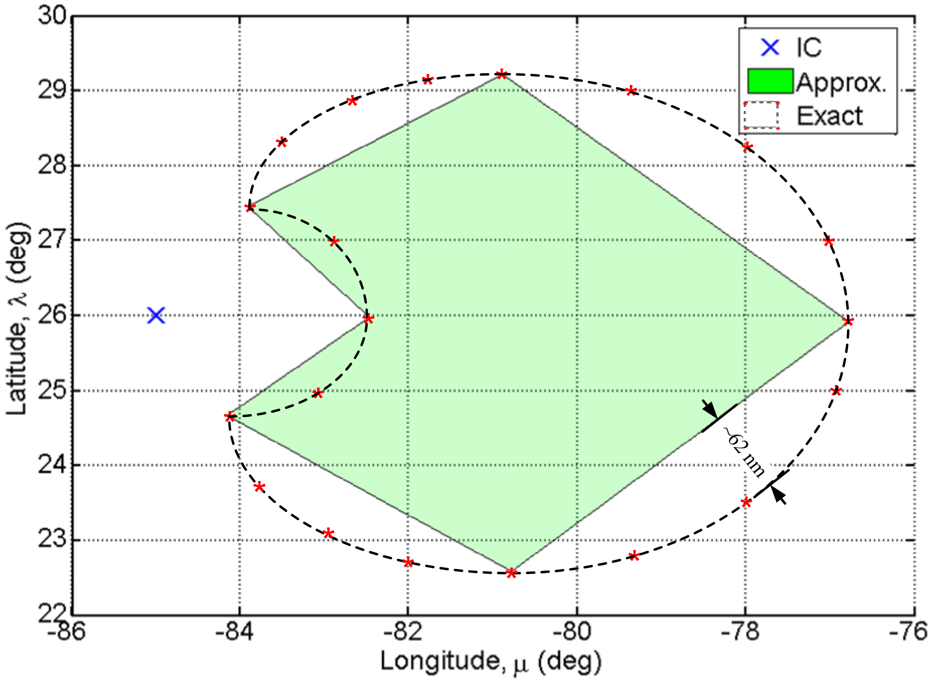


Figure 4.39 Accuracy of Polygonal Footprint Generation.

This is yet another example of how fidelity can affect solution accuracy and potentially risks the safety of the vehicle. On the other hand, computing a more accurate footprint requires more computational time; hence, faster processing speed and/or faster algorithms are required. Until improvements are made in either of these areas, having an approximate solution is better than having no solution at all; therefore, the near-term approach would be to implement a curve-fit on the lower-fidelity footprint that will provide a better estimate of the feasible region. Or, the polygonal approximations can be used to provide faster estimates of the infeasible regions. Consider the polygonal footprint given by Figure 4.36. By linearly connecting the maximum range positions, a region is "boxed-out" as illustrated in Figure 4.40. Any landing site within the polygonal footprint approximation, given by the green region, is a feasible option exemplified by the check-marked runway. Any landing site outside the blue-hatched boundary is infeasible exemplified by the x-marked runway. If the landing site is between the polygonal region and the "infeasibility boundary" then the feasibility is questionable, exemplified by the question-marked runway, unless additional trajectories are generated.

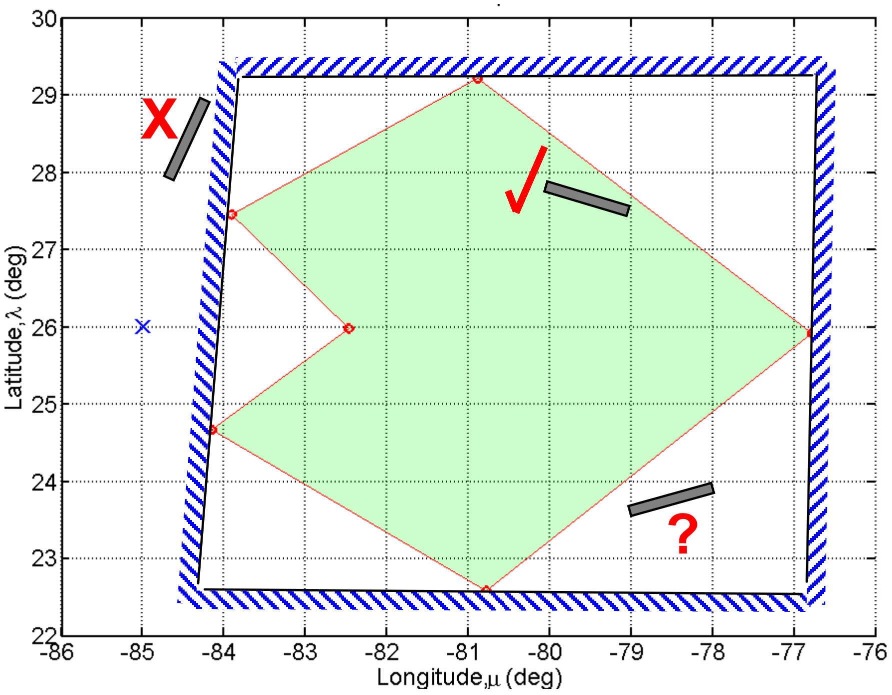


Figure 4.40 3-DOF Polygonal Footprint and Infeasible Region.

As illustrated, real-time optimal trajectory generation is critical for on-line planning and re-targeting scenarios. With online footprint determination, the vehicle's guidance system can now target any field within the feasible set.

e. Feasibility-Based Landing Site Retargeting Scenario

The following example demonstrates the proposed method's ability to rapidly generate optimal trajectories to various landing sites in Florida (FL). This example implements the FAC-generation logic to automatically generate the necessary end-point conditions required for safe handover to autoland guidance. By specifying the vehicle-specific requirement for minimum runway length, the FAC-generation algorithm can query a U.S. database of feasible landing sites. In this example, a database of potential FL sites was predetermined based on the requirement for a runway length of at least 10,000 ft. Table 4.8 lists the data for all runways that meets this requirement (data extracted from Ref. [180].

Table 4.8 Florida Airports with Runway Length Greater than 10,000 ft.

ID	Lat (deg)	Long (deg)	Elev. (ft)	Rwy #	Length (ft)	Width (ft)	Name
PAM	30.0700000	-85.57650000	18	13L/31R	10000	200	Tyndall AFB, Panama City
VPS	30.4833333	-86.52533330	87	12/30	12005	300	Eglin AFB, Valaparaiso
				1/19	10012	300	
JAX	30.4940556	-81.6878611	30	7/25	10000	150	Jacksonville Intn'l Airport
DAB	29.1799167	-81.0580556	34	7L/25R	10500	150	Daytona Beach Intn'l Airport
TTS	28.6150000	-80.6945000	10	15/33	15000	300	NASA Shuttle Landing Facility
XMR	28.4676667	-80.5668333	10	13/31	10000	200	Cape Canaveral AFS Skid Strip
MLB	28.1027528	-80.6452569	33	9R/27L	10181	150	Melbourne Intn'l Airport
MCO	28.4293925	-81.3089933	96	18L/36R	12005	200	Orlando Intn'l Airport
				18R/36L	12004	200	
				17R/35L	10000	150	
TPA	27.9754722	-82.5332500	26	18R/36L	11002	150	Tampa Intn'l Airport
PBI	26.6831606	-80.0955892	19	9L/27R	10008	150	Palm Beach Intn'l Airport
MIA	25.7932500	-80.2905556	8	9/27	13000	150	Miami Intn'l Airport
			8	8R/26L	10506	200	
RSW	26.5361667	-81.7551667	30	6/24	12000	150	Southwest FL Intn'l, Ft. Myers
HST	25.4884431	-80.3836658	7	5/23	11200	300	Homestead Air Reserve Base
NQX	24.5757014	-81.6888333	6	7/25	10000	200	Key West NAS

From this list, primary and secondary targets are selected based again primarily on runway length. Figure 4.41 shows the candidate FL runways with NASA

Shuttle Landing Facility (TTS) at KSC as primary (marked in red), Cape Canaveral Air Force Station Skid Strip (XMR), Eglin AFB (VPS), and Homestead Air Reserve (HST) as secondary (marked in blue), and the remaining runways are last resort (marked in black). Note that additional work can be done to intelligently automate the process of selecting runways. For example, there may be a critical tradeoff between runway length and distance of the site from the originally planned site.

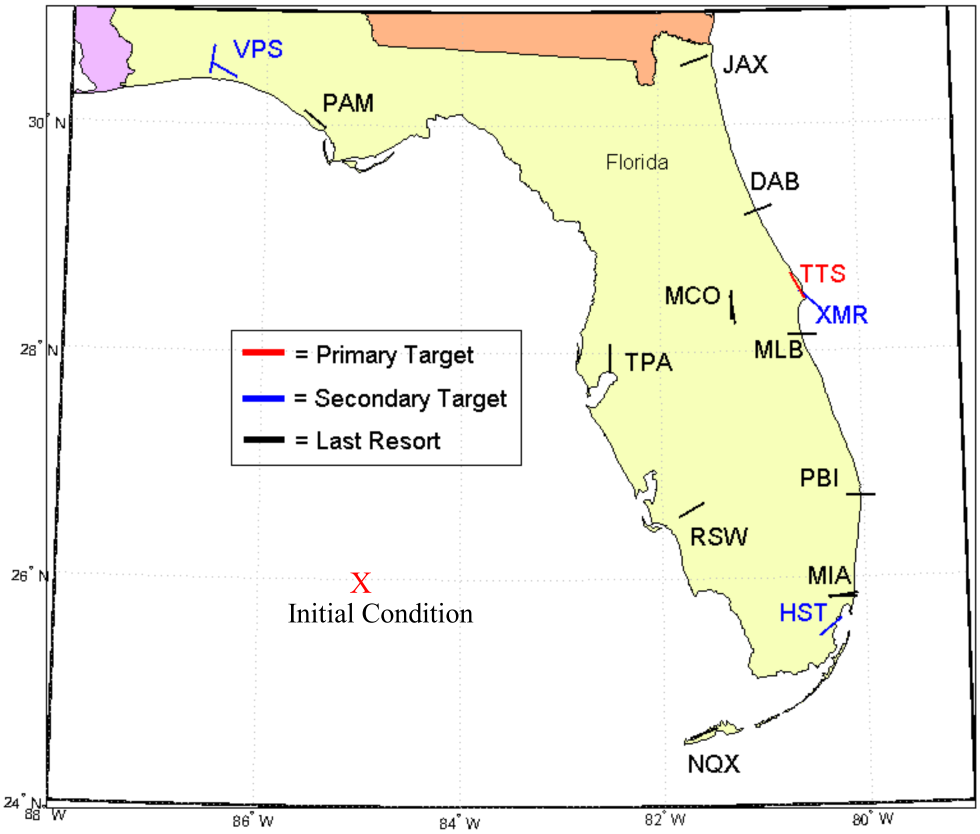


Figure 4.41 Terminal Guidance to FAC for Candidate FL Runways.

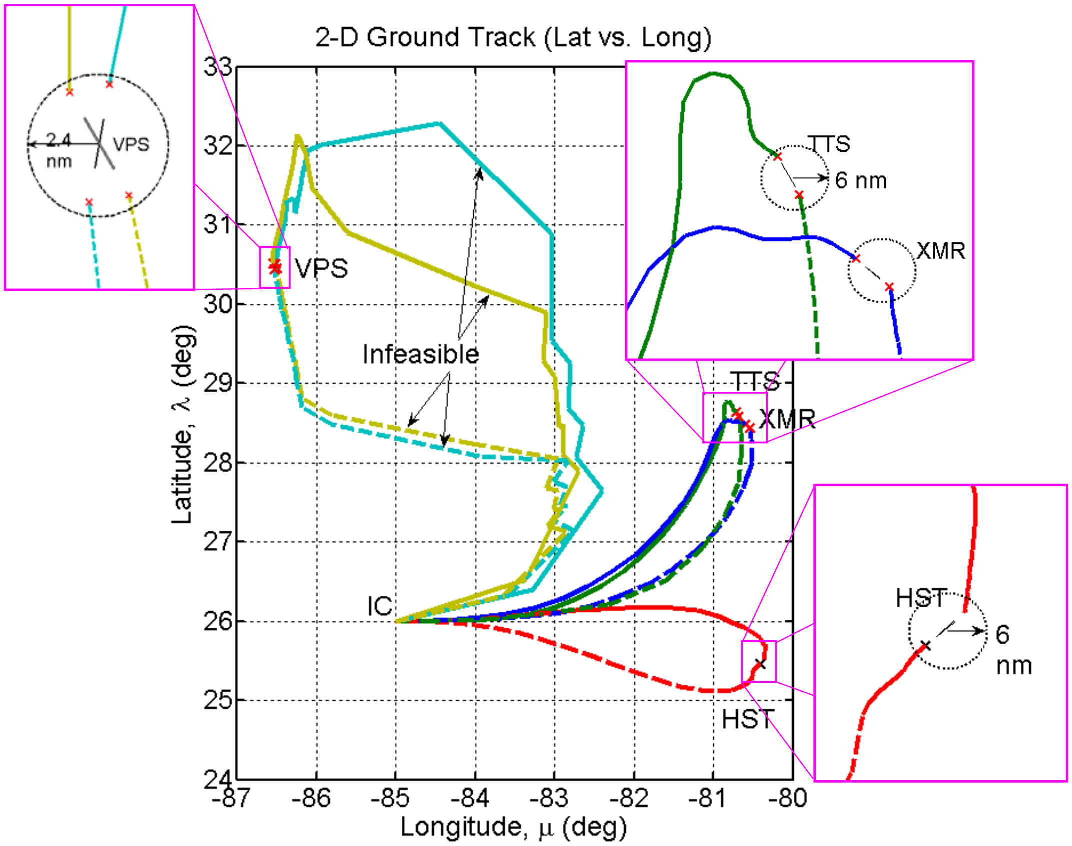


Figure 4.42 2D Ground Tracks for Trajectories Starting from IC to Primary and Secondary Targets.

The 2D ground tracks plotted in Figure 4.42, show that the trajectories to TTS, XMR, and HST are feasible, while those to VPS are not. If for some reason the vehicle could not land at TTS, it could still make an approach to landing at either XMR or HST provided the decision point coincided with, or was prior to, the initial condition (IC) mark. Notice that the feasibility was calculated for approaches from both ends of the runway. This is another feature built into the FAC-generation algorithm. Now, assume the onboard guidance system could generate max-range footprints fast enough. Overlaying the max-range footprints onto Figure 4.42, provides the information illustrated in Figure 4.43. This is the picture that the onboard computer (and/or pilots if manned) would see if such real-time footprint generation capability existed. Using the aero-tables, in this case, shrinks the landing footprint (i.e., reachable set) to the point that TTR and XMR are no longer reachable. Assuming that the actual tabulated aerodynamic data is perfect, using a crude aerodynamic approximation has potentially fatal results as indicated by the false positive of being able to reach TTS and XMR.

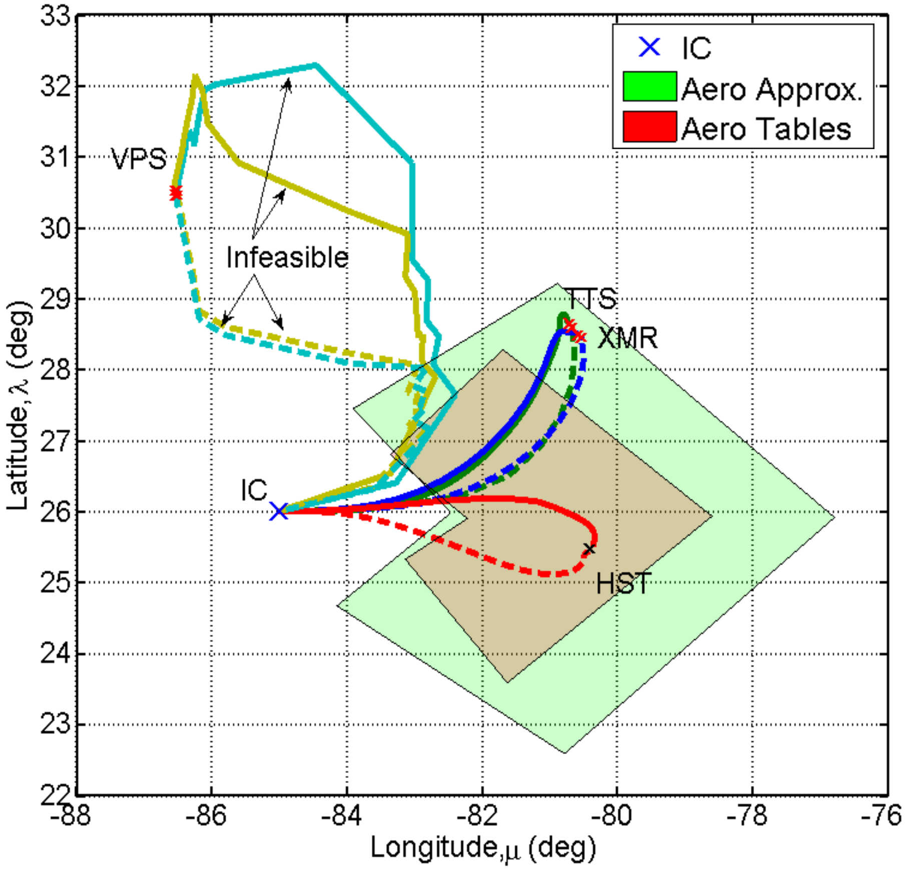


Figure 4.43 2D Ground Tracks for Trajectories Starting from IC to Primary and Secondary Targets with Overlayed Max-Range Footprints.

Ultimately, the accuracy of the footprint is highly dependent on the accuracy of the aerodynamic data. This effect implies that the approximation provides a condition where the L/D-ratio is higher, such as less drag. This enables a quick way of simulating the effects of a failed control, damaged body, or otherwise "dirty" configuration. In fact, some research efforts have scaled the drag coefficients that essentially simulates the effects of locked control surfaces [49],[181]. Since CPU-time of a curve-fitted aero-approximation is generally faster, a tradeoff now exists between speed and accuracy.

5. Numerical Considerations

The primary numerical consideration with the 3-DOF trajectory generation is computational time. As discussed in the motivation section of Chapter I, the GNC's ability to perform onboard, real-time computations is a critical aspect for autonomy / intelligence of unmanned vehicles. The less time it takes to compute high-accuracy, optimal trajectories, the more capabilities are provided. Since this research work did not attempt to optimize code for speed, there are a still lot of opportunities available as discussed in the future work chapter, Chapter VIII. One interesting observation worth further exploration is that there was some computational speed trade-offs when using equality or inequality constraints (i.e., events with epsilon-balls) for the final target conditions. From a more practical perspective, the actual position of the vehicle is governed by the accuracy of the sensors. As such, there should be an epsilon-ball that corresponds to sensor error. Although this work did involve some experimentation with epsilon-balls on initial and final conditions, the difference in results was considered irrelevant until implementing the feedback method.

C. CLOSED-LOOP, OPTIMAL TRAJECTORY GENERATION

1. 3-DOF Model

a. Vehicle

This section employs the same dynamical model as that used for the previous open-loop results. Although some open-loop runs were generated with the tabulated aerodynamic database provided by AFRL, the proceeding feedback implementation requires fast computations; hence, simulations use the aerodynamic approximation given by Eqs (4.4) and (4.5). It remains to be determined if the feedback approach works with the full table-look up model; nonetheless, as a first step in this new type of analysis, an analytical curve fit is implemented. Although the vehicle model is the same, feedback analysis requires the infusion of uncertainties and/or disturbances. As such, this work uses the inherent numerical propagation errors as the uncertainties and defines and integrates wind gusts into the system to simulate exogenous disturbances.

b. Wind Gust (Disturbance)

The most common form of an external disturbance encountered during atmospheric flight is the changing direction and intensity of the air mass itself (i.e., wind). For this work, a constant-magnitude wind gradient approximation is used for the disturbance model.

(1) Wind Gradient Approximation. As presented in Chapter III, typical wind disturbance analysis for air vehicles employs an analytical approximation to the wind's flow field characteristics. This may include the use of turbulence models and gradient approximations. This study employs a simplified approach to model a wind gust as illustrated in Figure 4.44(b).

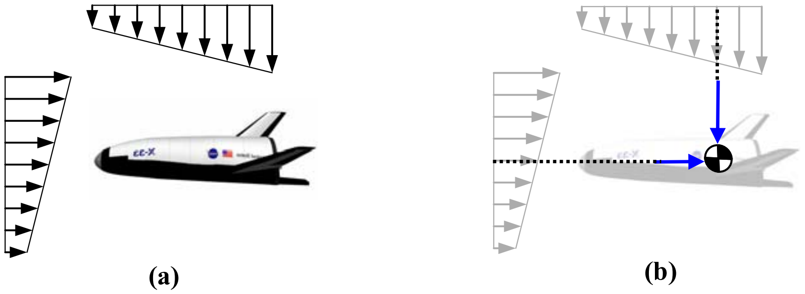


Figure 4.44 Approximation of Horizontal and Vertical Wind Gradients Acting on Vehicle.

Here, wind velocity components in the Cartesian x,y,z-directions $\left(W_x, W_y, W_z\right)$ are appended to the kinematical equations such that

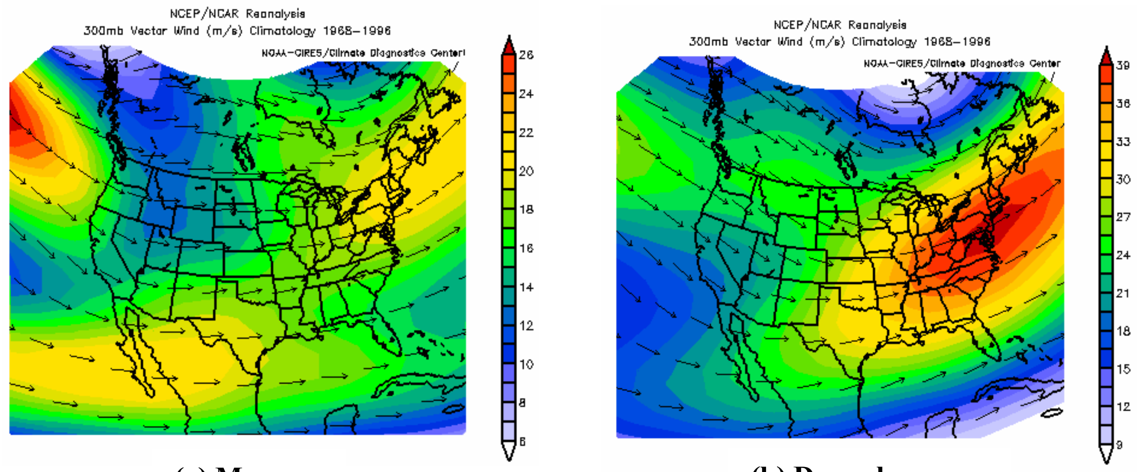
$$\begin{aligned} \dot{x} &= V\cos\gamma\cos\xi + W_x \\ \dot{y} &= V\cos\gamma\sin\xi + W_y \\ \dot{z} &= V\sin\gamma + W_z \end{aligned} \tag{4.42}$$

For the actual implementation, the wind coordinates are transformed from Cartesian to spherical wind components such that $[W_x, W_y, W_z] \rightarrow [W_\mu, W_\lambda, W_r]$. Also, details of the wind gradient are not included in the model, but rather uniform, constant components are used in the dynamical equations such that $\dot{W}$ terms are zero (i.e., steady winds). Now, with the wind modeled directly into the vehicle's kinematical equations, only the direction and magnitude of the representative wind gust is required. An important limitation of this simplified approach is that the wind only affects the vehicle's position and not it's velocity relative to the wind. For subsonic flight, large wind disturbances predominantly affect the vehicle's relative velocity and aerodynamic flight angles that consequently affect the aerodynamic forces and moments [26]. Despite that these effects are omitted and only the effects on position are considered, this approach still provides a preliminary study for testing the robustness of off-nominal conditions via exogenous disturbances.

Although arbitrarily selecting winds will suffice for this study, it is common that disturbance rejection analysis incorporates a worse case scenario in order to test the robustness of the proposed design.

(2) Worse-Case Winds. To study the effects of wind on atmospheric flight vehicles, it is important to consider a worst-case scenario. A "nominal" wind depends on a particular region of the atmosphere, current temperature and pressure conditions, and geographical location. Unfortunately, for a vehicle transcending large vertical and horizontal distances through the atmosphere, many wind uncertainties exist. Although historical data can be used to chart trends, sudden and local variations in wind direction and intensity (e.g., turbulence and wind shear) make wind prediction nearly impossible. Because of these uncertainties and unpredictability, it is important to test for the worse possible case and in most design applications, to even add a margin of safety by exaggerating the worse-case.

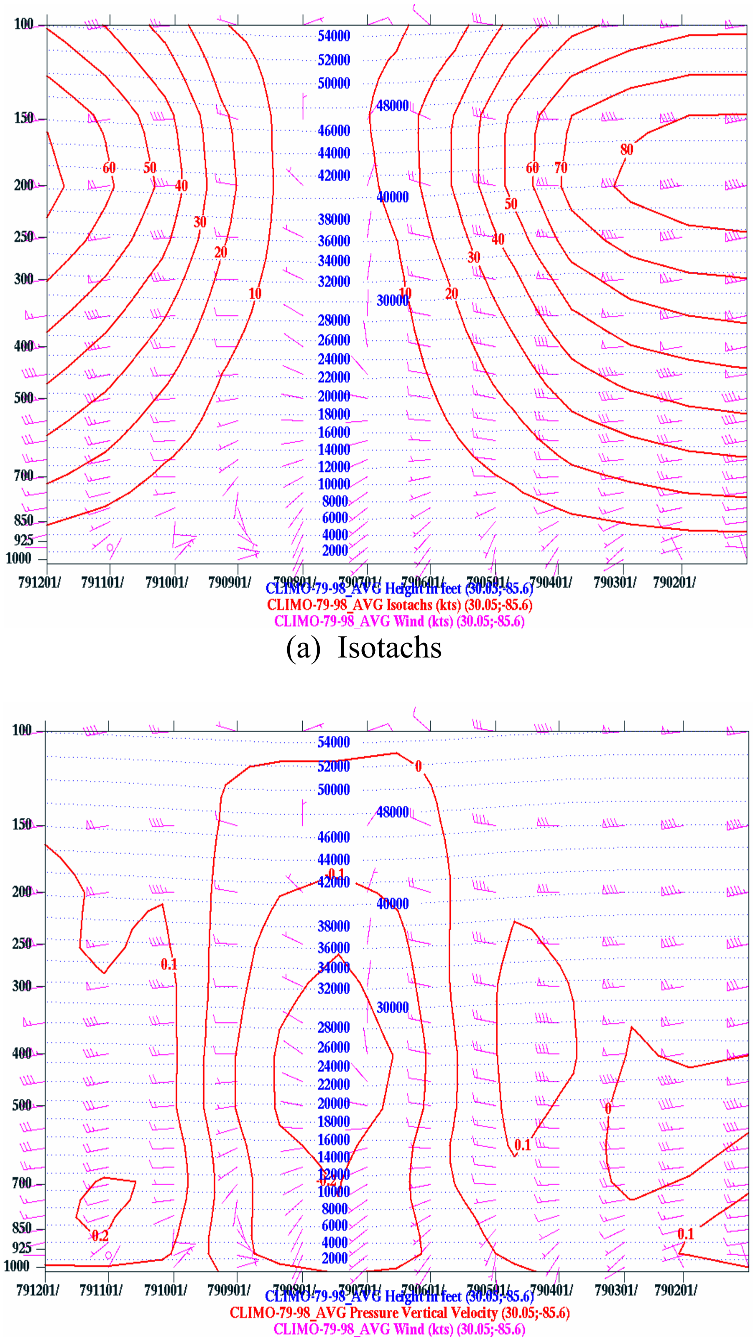
Upper air winds, or winds aloft, over the continental U.S. typically flow from west to east as illustrated by the "westerly" velocity vectors in Figure 4.45 showing long-term means (1968-1996) for May (a) and December (b). The troposphere, extending from the surface of the Earth to about 5-9 miles (26,400-47,520 ft) in elevation is the densest region of the atmosphere where most weather phenomena occurs [182]. It is the high upper region of the troposphere where the jet stream velocity can reach approximately 80 to 180 mph heading east.


NCEP/NCAR Reanalysis
300mb Vector Wind (m/s) Climatology 1968–1996
NOAA–CIRES/Climate Diagnostics Center
26
24
22
20
18
16
14
12
10
8
6
NCEP/NCAR Reanalysis
300mb Vector Wind (m/s) Climatology 1968–1996
NOAA–CIRES/Climate Diagnostics Center
39
36
33
30
27
24
21
18
15
12
9

(a) May **(b) December**

Figure 4.45 Example of Wind Flow Patterns over Continental U.S. at 300mb (~30,000 ft) [183].

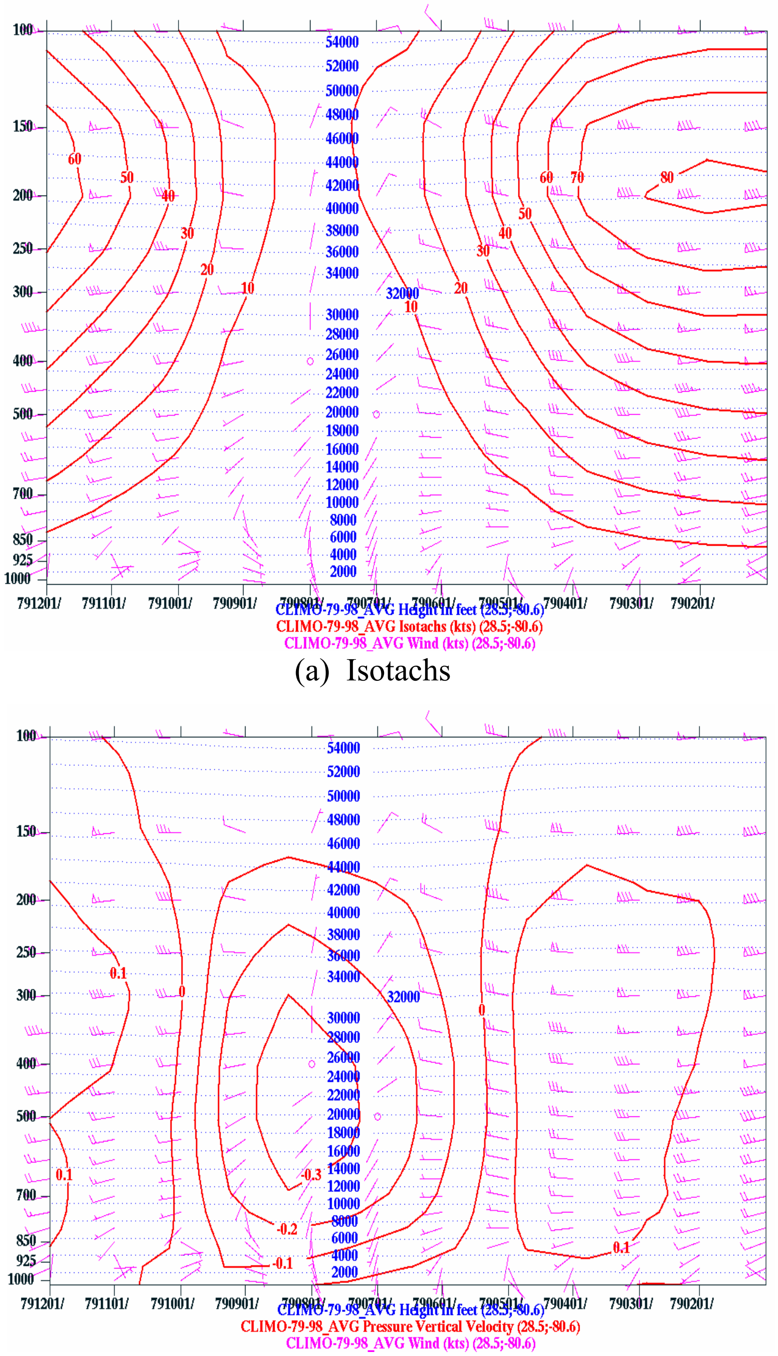
The wind components used in this demonstration represent a south-easterly flow which corresponds to a north-westward heading direction with the addition of a large vertical downward component. For winds aloft, a downward vertical component of wind is typically very small; on the order of 1-3 cm/s [184]. In localized storm systems this vertical component may exceed 25 m/s (> 82 ft/s) [185]. For the continental U.S., this direction and magnitude of flow is commonly encountered in hurricane-force winds typically in the vicinity of the Gulf of Mexico. In this worse case scenario, a microburst or significant wind shear in the form of a column of sinking air is capable of generating wind speeds higher than 160 mph [186]. Since vertical windshears mostly occur at altitudes below 1000 ft, they are more commonly encountered during takeoffs and landings; however, there are cases of large vertical winds at higher altitudes. A relevant Space Shuttle example is the uncommon but problematic occurrence of large vertical wind shears during interactions between the Polar Jet and the Sub-Tropical Jet near the base of the jet stream (~ 25,000 to 35,000 ft) over the south-eastern U.S.. Figure 4.46 shows the climatology over Panama City based on averaged data from 1979 to 1998. Likewise, Figure 4.47 shows the climatology over Cape Canaveral. As seen in both, there is an appreciable amount of wind velocity component in the vertical direction.



(a) Isotachs

(b) Pressure Vertical Velocity

Figure 4.46 Climatology over Panama City, FL (1979-1998) Showing Averaged Annual Height (ft) and Wind Speed (kts) [187] .



(a) Isotachs

(b) Pressure Vertical Velocity

Figure 4.47 Climatology over Cape Canaveral, FL (1979-1998) Showing Averaged Annual Height (ft) and Wind Speed (kts) [187].

(5) Wind Vector Selection. Since large vertical wind shear is problematic near the Jet Stream, the wind vector for this study has a substantial vertical component. For the given initial conditions and final conditions corresponding to landing at TTS (KSC), FL, the wind is applied over a period of 20 sec beginning 460 sec into the flight. This time-of-flight duration (460 to 480 sec) corresponds to an altitude of approximately 45,000 to 30,000 ft, overlapping the high upper troposphere where the Jet Stream reaches its peak. Also, this altitude corresponds to the region of transition (tropopause) from the upper atmosphere (stratosphere) to the lower atmosphere (troposphere) where a large wind shear phenomenon is likely to occur.

Based on averaged wind data from 1979 to 1998 climatology charts for the south-eastern U.S. region [187], a nominal wind magnitude of about 30 kts is appropriate, so for this study, a magnitude of 33.25 kts was selected as the initial "light" wind, see Table 4.9. For trajectory analysis, Table 4.9 shows the wind gusts selected for comparisons to a no wind scenario.

Table 4.9 Wind Gust Selection for Trajectory Comparisons

	Wind Components, fps			Wind Magnitude		
Type	Wx	Wy	Wz	fps	kts	mph
Light	-30.00	15.00	-45.00	56.12	33.25	38.27
Moderate	-67.50	33.75	-101.25	126.28	74.82	86.10
Severe	-101.25	50.63	-151.81	189.37	112.20	129.12
Exaggerated	-151.88	75.94	-227.81	284.13	168.34	193.73

Based on the Saffir-Simpson Hurricane Scale for wind speed categories [188], the worse-case scenario, labeled "exaggerated" in Table 4.9, corresponds to wind gust intensity in the "Cat 5" hurricane range (> 155 mph). Note that although it is labeled "exaggerated" in this study, it is not unrealistic. For example, Figure 4.48 shows Hurricane Katrina over the Gulf of Mexico and Figure 4.49 shows a sample of Katrina's velocity vector chart at 40,000 ft at a measured 10 m/s (32.8 ft/s). Note the direction of the vectors are consistent with the vector directions used in this study. As discussed later, results show that the vehicle's flight guidance algorithm can tolerate this significant wind gust intensity.

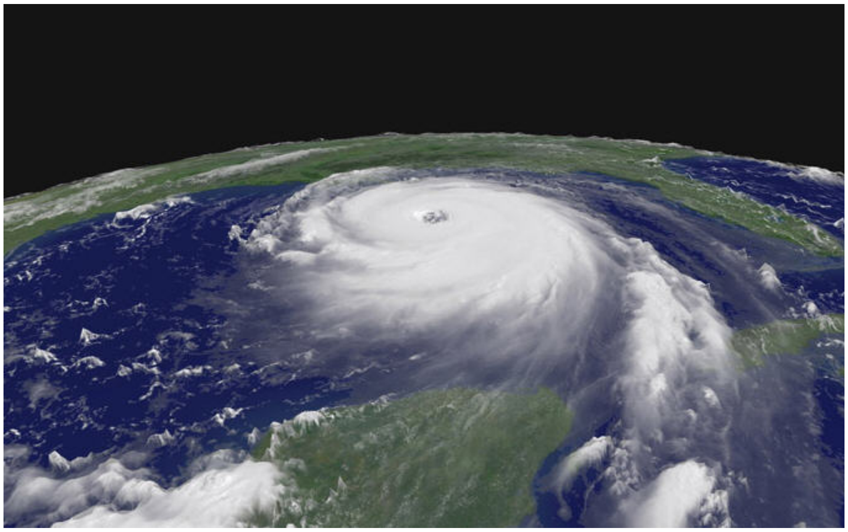


Figure 4.48 Hurricane Katrina Over the Gulf of Mexico (Courtesy of NOAA) [189].

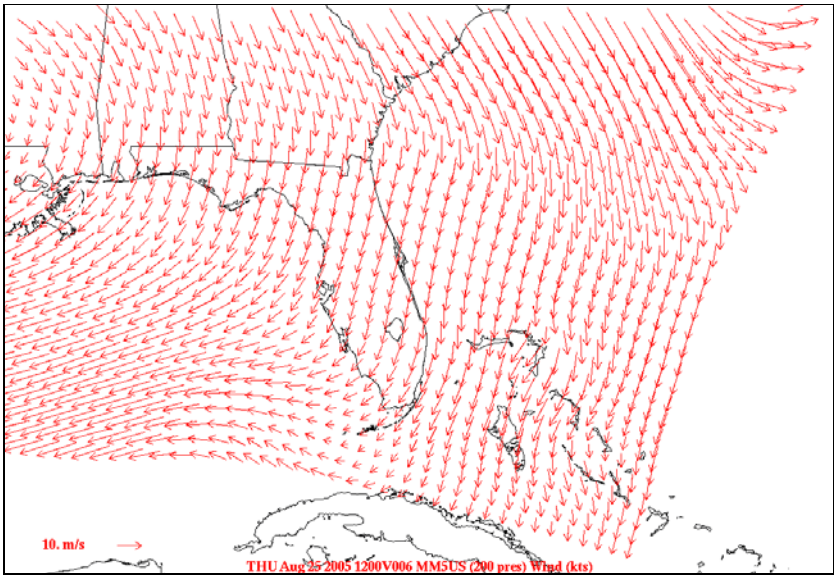


Figure 4.49 Hurricane Katrina Wind Velocity Vector Chart at 40,000 ft [190].

Remarks: Provided by Brown, Figure 4.49 was obtained from the Air Force Weather Agency (AFWA) version of a mesoscale model using AFWA's 15-km CONUS window [184].

(6) Determination of Maximum Wind (Disturbance Rejection Performance). With testing any control algorithm, it is important to analyze the controller's disturbance rejection performance and estimate what maximum disturbance the feedback system can tolerate until the system in no longer controllable. For this study, the system's performance is numerically tested. In order to determine how much wind the vehicle can handle before the optimal feedback guidance algorithm fails to converge, various wind profiles are implemented with increasing magnitude. For purpose of comparison, the wind direction is fixed by using the same direction cosine (i.e., same unit vector) and scaling the velocity magnitude, at most, by 50% increments. Increasing the winds until problems occur (e.g., constraint violations or non-convergence) helps indicate the maximum allowable wind-gust magnitude.

2. Problem Formulation

The optimal control problem is to find the control history that minimizes the miss distance to the center of the FAC. This closed-loop implementation uses an epsilon-ball on the final conditions. In this case, this epsilon is based on the 3D-cube geometry of the FAC as previously defined in Figure 4.3. Now, with the final conditions corresponding to a landing approach at the Shuttle Landing Facility (TTS, FL), the following boundary conditions are specified

$$\begin{aligned}
&(t_0, h_0, \mu_0, \lambda_0, V_0, \gamma_0, \xi_0) = (0\,\text{sec}, 167323\,\text{ft}, -85^\circ, 30^\circ, 8530.2\,\text{ft/s}, -1.5^\circ, 0^\circ) \\
&(h_f, \mu_f, \lambda_f, V_f, \gamma_f, \xi_f) \in [(2000 \pm 400\,\text{ft}, -80.7112^\circ \pm 0.001371^\circ, \\
&\qquad\qquad 30^\circ \pm 0.001097^\circ, 300\,\text{ft/s}, -6.0^\circ, -60^\circ)]
\end{aligned} \tag{4.43}$$

Except for the changes with the initial and final conditions given by Eq. (4.43), all other constraints and model parameters are the same for the closed-loop problem formulation.

3. Feedback Implementation

In recent years, it has been shown using relatively high-fidelity models that PS methods are capable of generating real-time trajectories for reentry vehicles [83]. Combining this capability with a PS feedback implementation, based on new theoretical foundations, provides a means for real-time optimal feedback. For entry guidance, this feedback approach updates the guidance laws in a Carathèodory-π sense, as described in Sec. II. This type of an update permits a longer computational time that may be exploited to solve high-fidelity trajectory optimization problems in real time.

4. Results and Analysis

a. Open-Loop Solution

For the reentry trajectory, the open-loop terminal guidance solution takes only about 30 seconds to generate from an arbitrary starting point. Although 30 seconds may be viewed as rapid in the context of "setup time," this computation time can be reduced by at least a factor of 100 by optimizing the actual code, eliminating the Windows and MATLAB overhead etc. [151]. None of these computational enhancements were carried out because the purpose of this current work is to demonstrate the principles. In any case, Figure 4.50 and Figure 4.51 show that the optimal open-loop control given by the α, σ-rate modulation plotted in Figure 4.52 drives the RLV to the FAC-target box over a total flight time of 669.1 seconds and within the allowable tolerance for the desired end-point conditions.

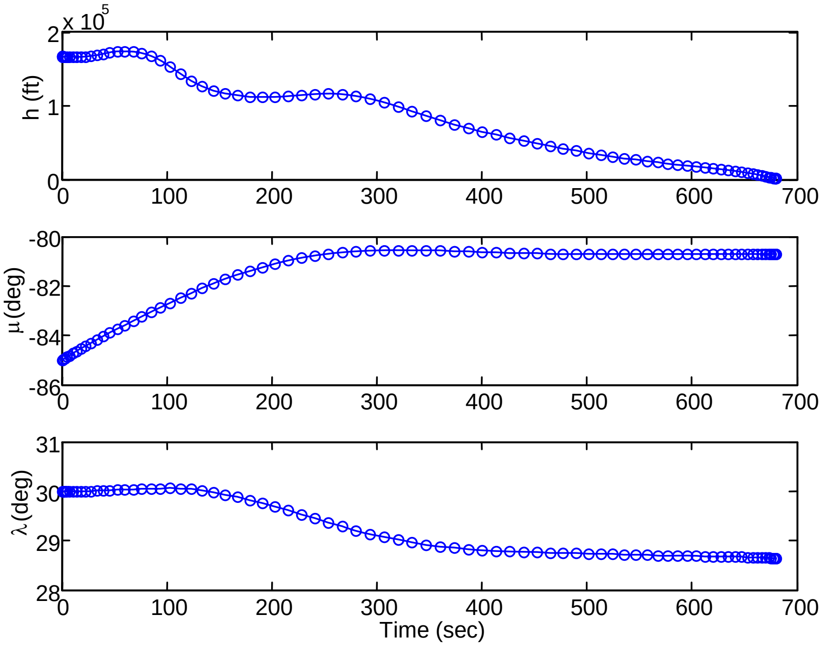


Figure 4.50 PS-Computed Open-Loop Optimal States (Position).

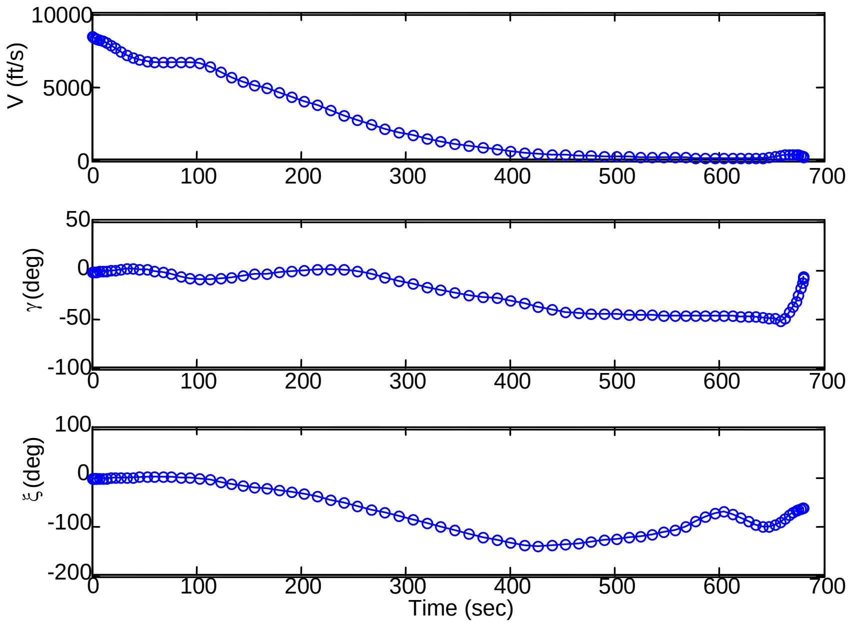


Figure 4.51 PS-Computed Open-Loop Optimal States (Velocity).

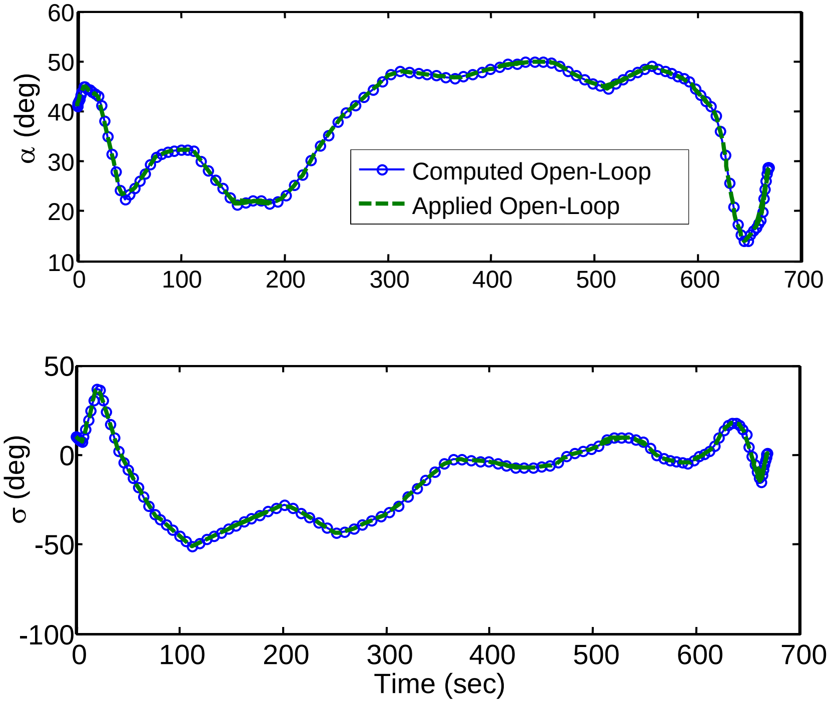


Figure 4.52 PS-Computed Open-Loop Optimal Controls.

Note that Figure 4.52 shows the optimal open-loop controls computed and applied to the 3-DOF RLV model. The difference being that the solution computed is solely based on the DIDO trajectory (data points coincide with number of nodes) whereas the applied control is the DIDO-interpolated result that is applied to the plant dynamics. Figure 4.53 shows the resulting 3D flight trajectory after applying the control sequence.

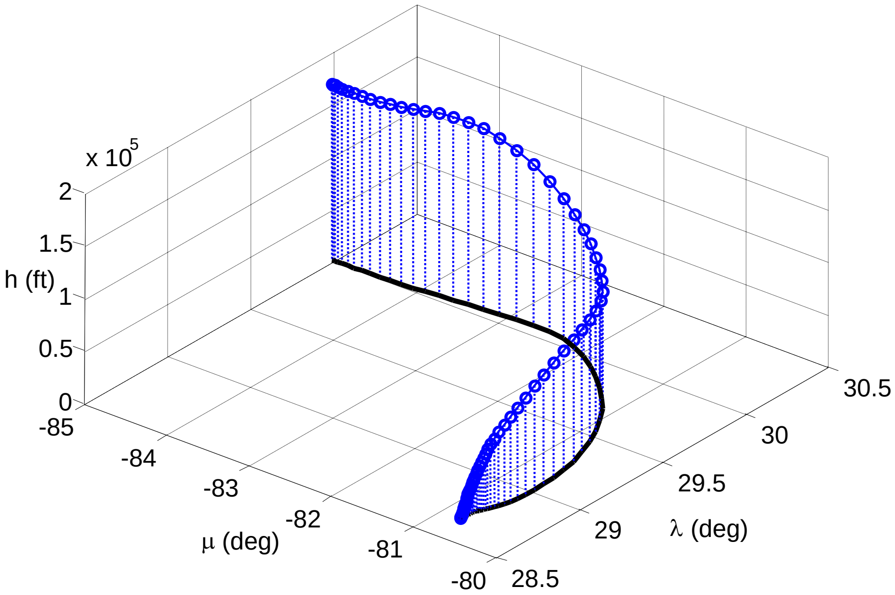


Figure 4.53 PS-Computed Open-Loop Optimal 3D Flight Trajectory.

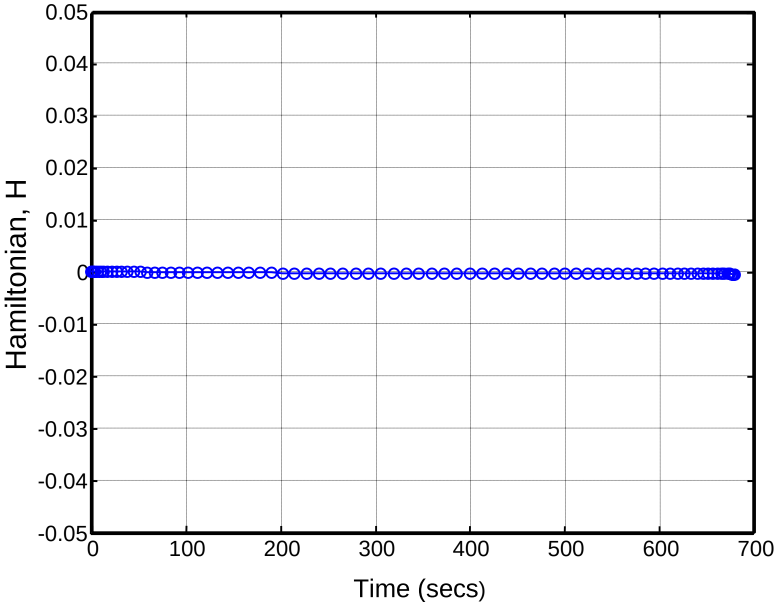


Figure 4.54 Open-loop Hamiltonian Evolution.

The controls are optimal in the sense that they satisfy all the necessary conditions for optimality; hence, strictly speaking they are only extremals. Again, combining the HVC and the HEE conditions, the Hamiltonian is zero for all time, clearly evident by Figure 4.54. Although not included here, the open-loop system response does satisfy the endpoint conditions within an acceptable error range.

b. Closed-Loop Solution

Although the previous open-loop solution results in a feasible trajectory and desired endpoint conditions that satisfy the necessary conditions for optimality, it does not account for any external disturbances as would be the case in real applications. To illustrate the effects of external disturbances for this reentry problem, a simulated wind gust was applied over a period of 20 seconds beginning at 460 seconds into the flight. Sensor measurement errors and parameter uncertainty can be simulated by assuming the role of the errors from the numerical propagation. The effects of the wind on the open-loop solution are seen in Figure 4.55 to Figure 4.57.

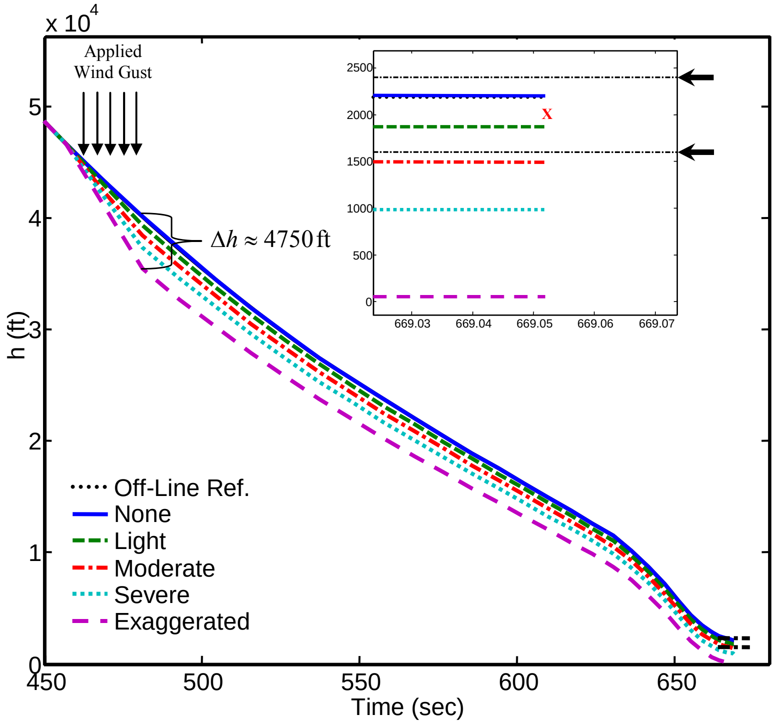


Figure 4.55 Effect of Wind Gusts on Altitude for Open-Loop System.

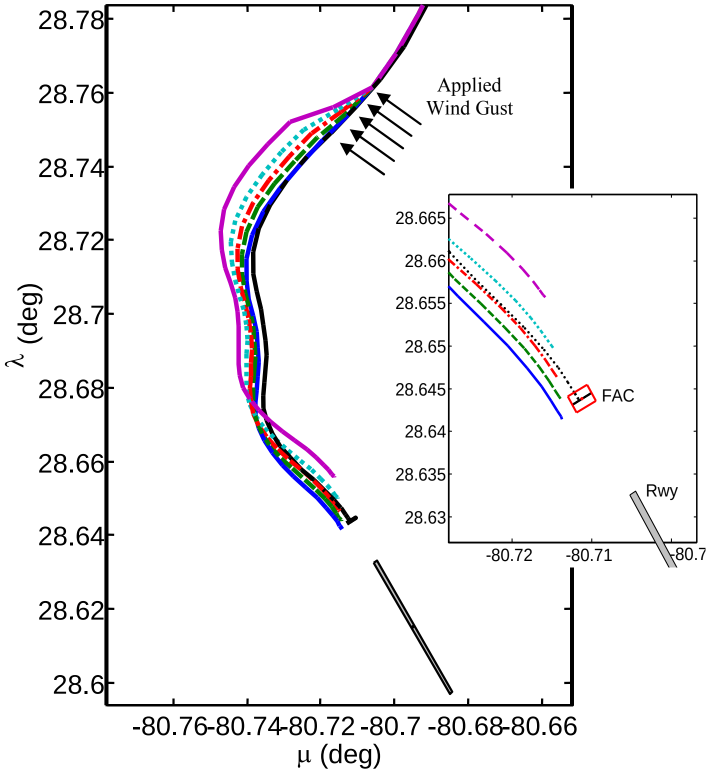


Figure 4.56 Effect of Wind Gusts on Position for Open-Loop System.

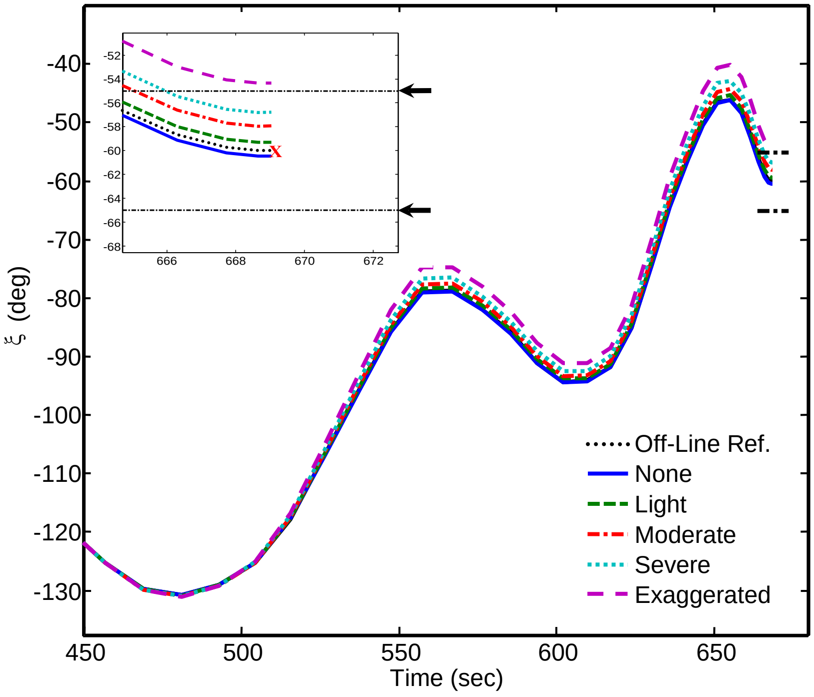


Figure 4.57 Effect of Wind Gusts on Heading for Open-Loop System.

Increasing the wind-gust intensity pushes the vehicle further from the intended target as indicated by the “x”-mark in the insets. Figure 4.55 shows that an “exaggerated” wind-gust (i.e., > Cat 5 hurricane) causes a rapid decrease in altitude of almost 5000 ft. Figure 4.57 shows that even with wind gusts up to and including “severe,” the final heading is still within limits as indicated by arrow-marks on the inset. Some errors and disturbances are acceptable because the FAC-target design allows margin on the vehicle’s final conditions; however, with large disturbances such as the “exaggerated” wind gust, the system is not capable of compensating without feedback. The 2D ground track in Figure 4.56 is a good example of how other errors in the system (e.g., modeling, numerical propagation, etc) can negatively affect the trajectory. Here, even the no-wind case fails to hit the FAC-target box. It is evident that feedback is required to handle these uncertainties and disturbances.

With the open-loop solution now generated, the closed-loop system is initiated by using the open-loop solution as a “start-up” guess. Figure 4.58 and Figure 4.59 shows the effects of wind gusts on the closed-loop trajectory for the variables of interest. In contrast to the open-loop results, it is clear that all the closed-loop trajectories satisfy the final conditions, including the “exaggerated” case with a wind-gust magnitude of 194 mph! Also note how the 2D ground track nicely converges to the center of the FAC-target in Figure 4.59. As a reminder, the vehicle is not guided along a reference trajectory. In fact, the concept of reference trajectory tracking is abandoned altogether; rather, the principle of autonomous real-time trajectory generation is adopted.

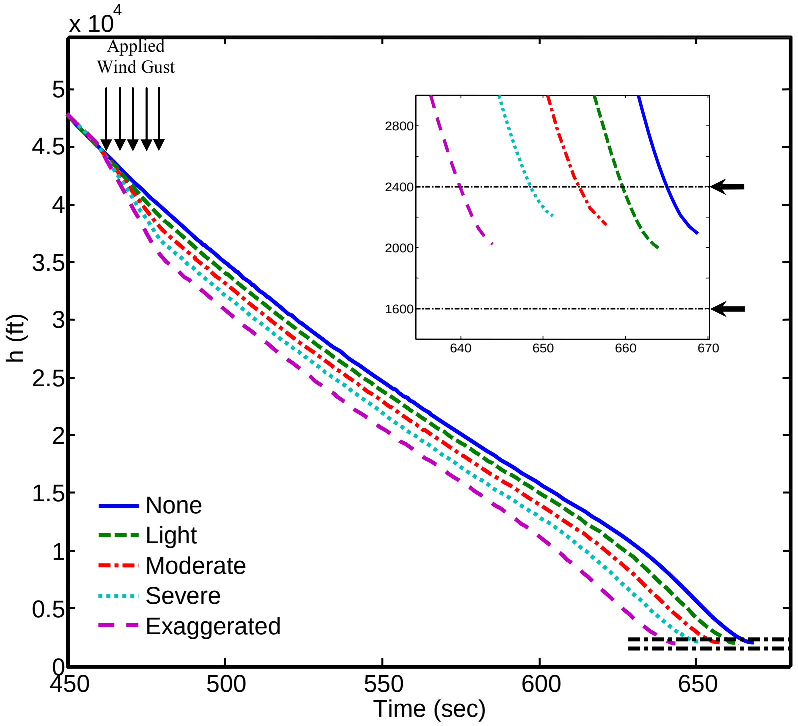


Figure 4.58 Effect of Wind Gusts on Altitude for Closed-Loop System

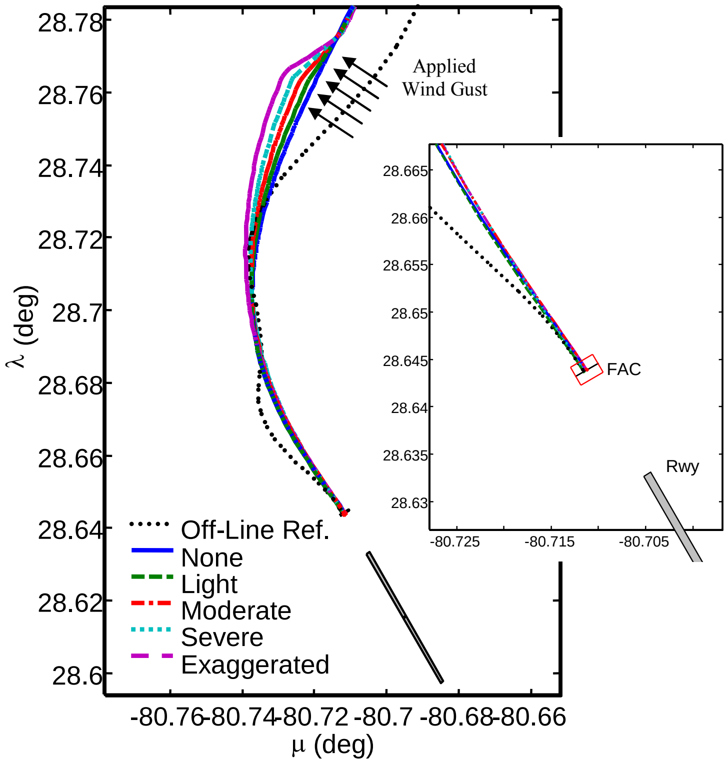


Figure 4.59 Effect of Wind Gusts on Position for Open-Loop System

As expected, the open-loop controls drive the model to the target conditions under nominal conditions but the simulation with large external disturbances fails to meet the requirements (i.e., end-point conditions defined by the FAC).

The closed-loop solution does present what appears to be signs of control "chatter" as indicated in Figure 4.60; however, note that this is not chattering but rather an artifice of compressed x-axis scaling. In this case, none of the "spikes" exceed a realistic rate of 5 deg/s. This noticeable oscillatory behavior in this region is a result of two potential situations specific to the generated trajectory. First, the open-loop control sequence is near saturation in this region which would imply that demanding more control will cause "chatter." Second, around 500 sec into the close-loop implementation, the computational time increases slightly as a result of less nodes being used in the successive "guess" structure of the feedback algorithm. This causes an additional time delay that ultimately affects the control.

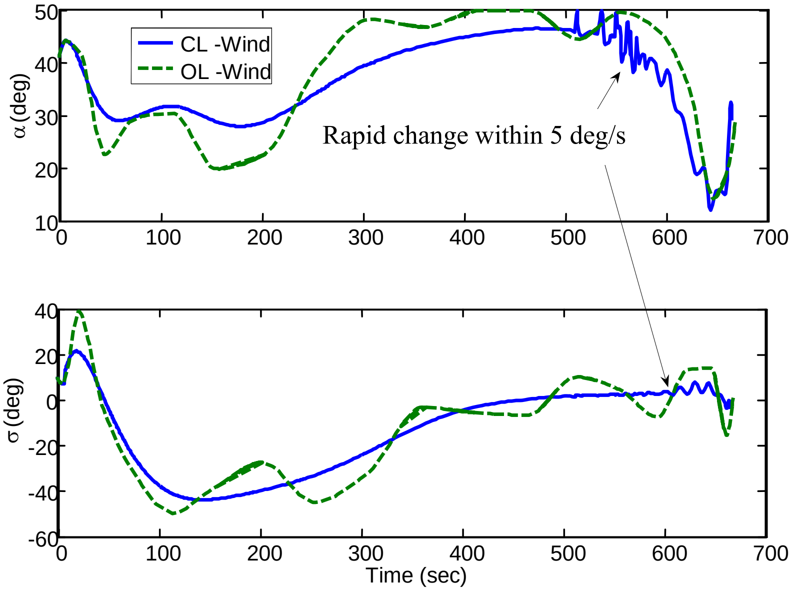


Figure 4.60 Example of Control Oscillations for Closed-Loop System.

In order to demonstrate the principles described in the Modeling section, the effect of limiting the control rates by imposing rate constraints is investigated. Figure 4.61 shows an example of a similar scenario as in Figure 4.60, but without experiencing any rapid control oscillations; hence, demonstrating the smoothing effect of pseudo-controls.

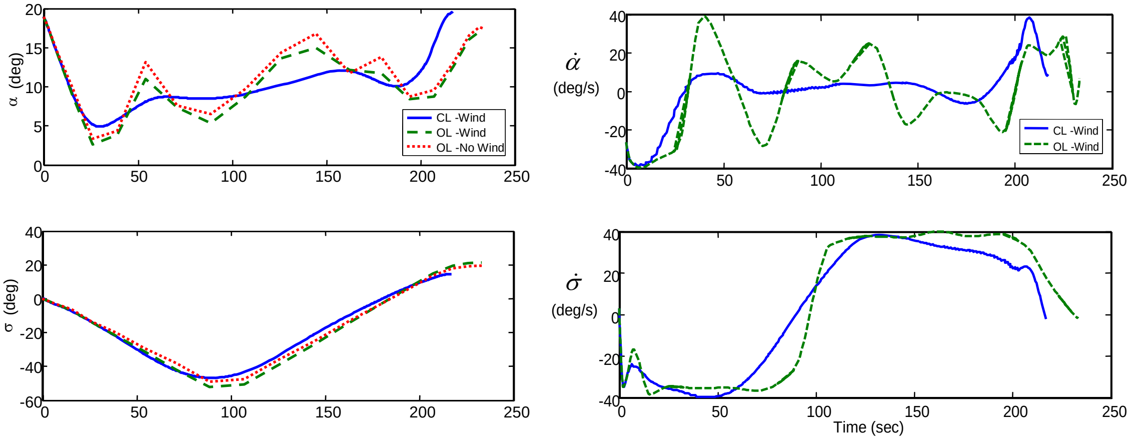


Figure 4.61 Example of Modeling Rates as “Virtual” Controls to Smooth Control Oscillations.

Results from a similar model, but without path constraints, reveals that a wind gust of approximately 206 ft/sec (~ 140 mph) will cause the vehicle to impact the earth’s surface unless the closed-loop algorithm is implemented. Shown in Figure 4.62, the vehicle flying the open-loop trajectory is slammed into the ground, whereas the closed-loop trajectory corrects for the microburst and is able to achieve a final altitude of 1629 ft, within the FAC-box vertical limits of $2000 \pm 400\,\text{ft}$.

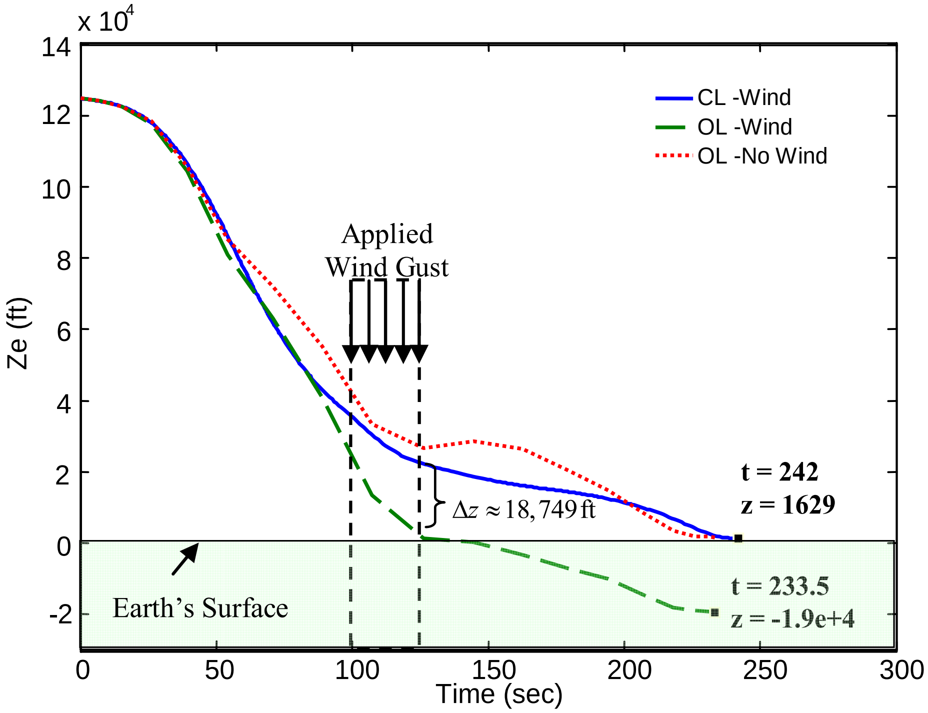


Figure 4.62 Worse-Case Scenario: "Exaggerated" Wind Gust Causing Earth Impact.

Both cases (Figure 4.58 and Figure 4.62) demonstrate that a viable trajectory exists under hurricane-force wind gusts when implementing the optimal nonlinear feedback. Also, like the case for no disturbances, it is important to note that for small disturbances, such as the light-wind scenario, the closed-loop trajectory approximately tracks the off-line open-loop reference trajectory. This is a direct result of Bellman's Principle. Since the initial condition to the re-optimization problem lies approximately on the original optimal path, then the new solution will be along the same trajectory. This "stability" result is a direct consequence of optimality [191].

As with all reentry problems, it is important to ensure that the vehicle does not exceed limits on dynamic pressure, heating, and structural loads. A benefit of this constrained optimal feedback guidance is that the path constraints are not violated as demonstrated by Figure 4.63 and Figure 4.64.

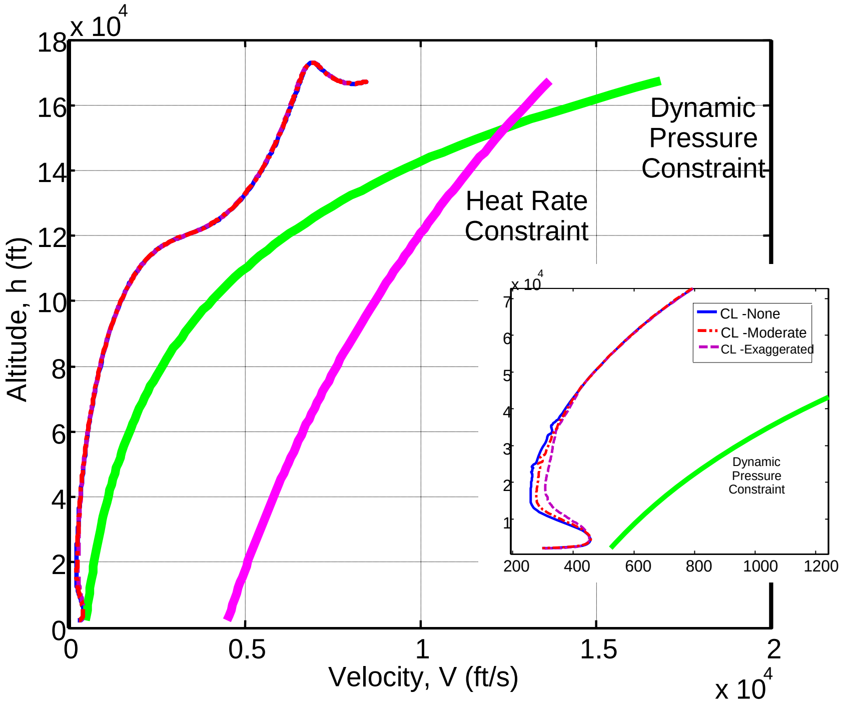


Figure 4.63 Velocity vs. Altitude and Path Constraints with Wind Disturbance.

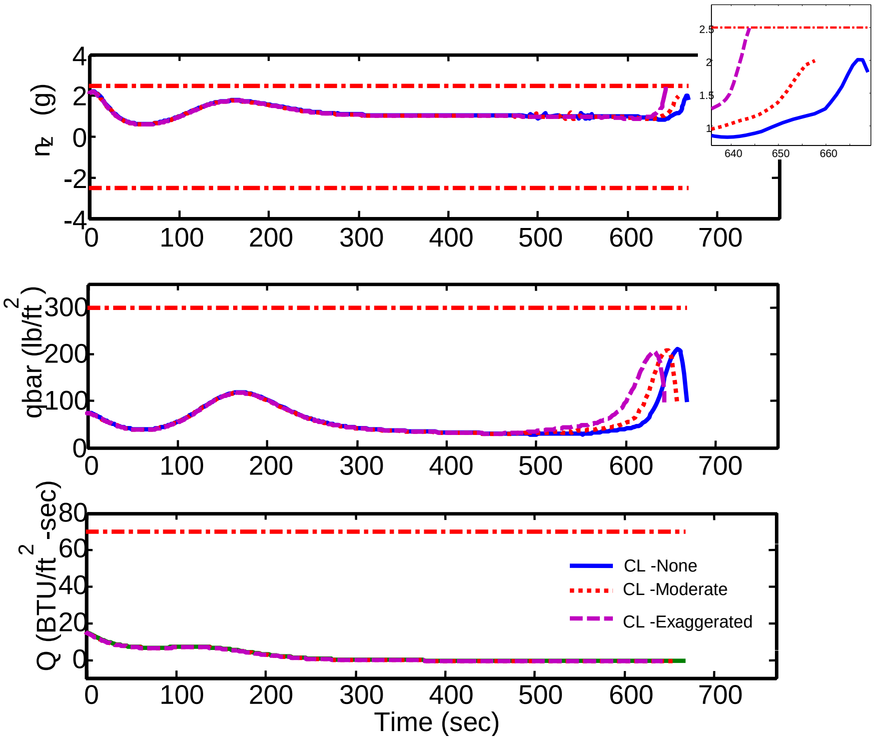


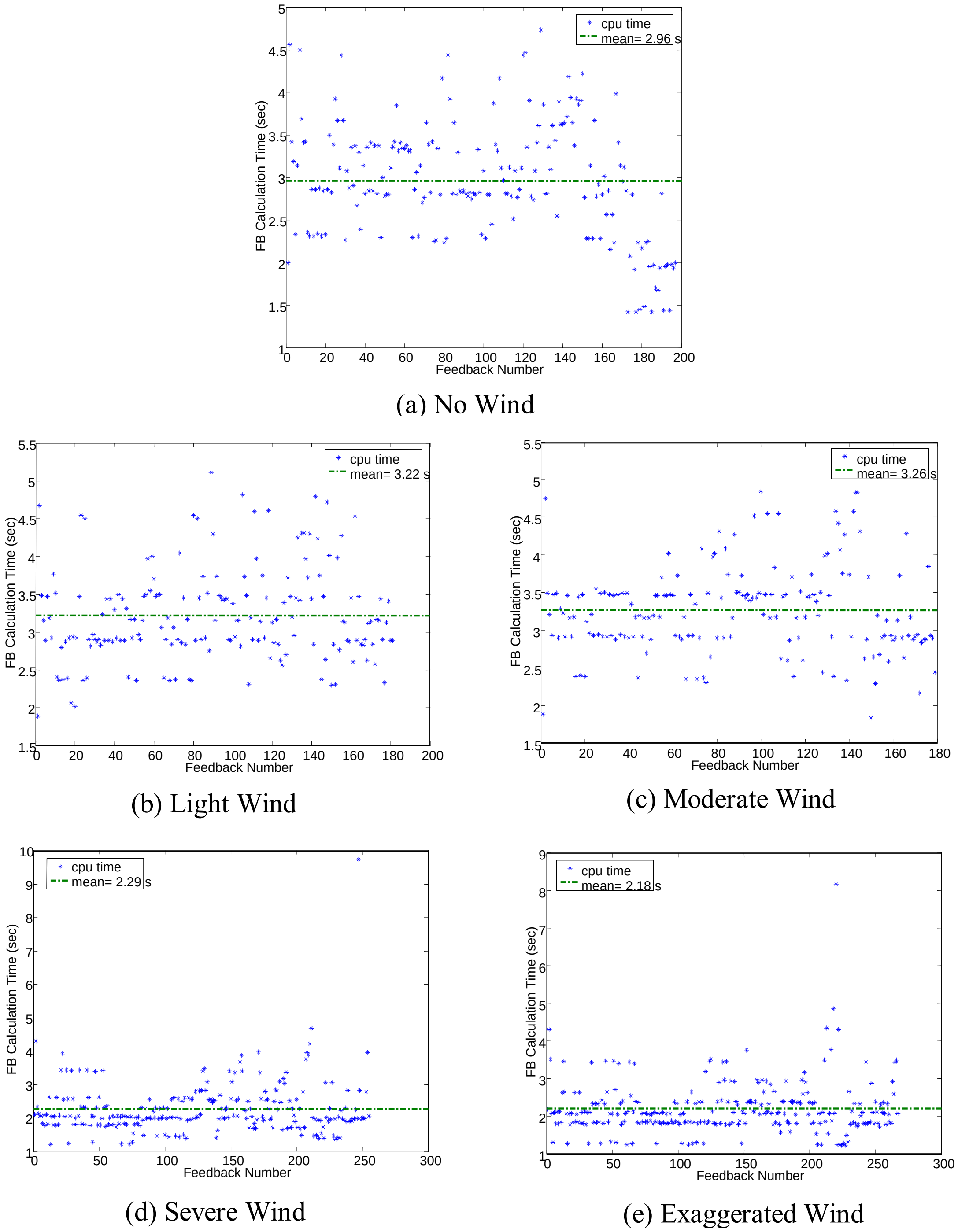
Figure 4.64 Comparison of Path Function Time Histories.

Depending on the specific initial conditions, the resulting flight path may or may not have active path constraints. Figure 4.63 shows how a sample of closed-loop trajectories do not cross the plotted dynamic pressure and heat rate constraints and also have a margin of safety. Note that this margin of safety can also be explicitly enforced via a modification to the cost function.

Overall, the RLV vehicle can handle wind gusts approximately 7.7 % times its current air speed. For this study, a vehicle traveling from 45,000-30,000 ft with a corresponding airspeed of 2650-1560 ft/s, can tolerate a maximum wind gust magnitude of approximately 205 ft/s (~ 140 mph).

c. Computational Time

Success of this feedback method depends heavily on relatively fast computation time. A theorem that links the required computation time to the dynamics of the system was summarized in Chapter II. Computation of the first off-line, open-loop optimal trajectory takes an *average* of 12 seconds and the subsequent open-loop optimal control updates are computed within approximately 1.2 to 9.75 seconds depending on the disturbance induced by the wind gusts, and the stability of the Windows environment at the time of run. These trajectories were generated on a Dell Optiplex Desktop computer with a Pentium M, 3.40 Ghz processor, and 1.0 GB of RAM. The feedback computation times are shown over the entire trajectory in Figure 4.65(a)-(e).



(a) No Wind

(b) Light Wind

(c) Moderate Wind

(d) Severe Wind

(e) Exaggerated Wind

Figure 4.65 Feedback Computation Times for Closed-Loop System.

For all wind disturbance cases, the average CPU computation time was 2.78 sec. As noted earlier, it is possible to reduce this time by a factor of 100. Nonetheless, 2.78 sec is quite adequate for declaring the computational speed as "real-time" as defined by the theory developed in Ref. [148] (see Chapter II) and validated by the numerical experiments reported in this section.

d. Comparing Zero Order Hold Method

This section tests the theoretical premise that this PS-feedback implementation is superior to the traditional sample-and-hold type methods as discussed in Chapter II. The following series of figures compare the clock-based PS-feedback controls used in the previous section to a zero-order hold (ZOH) method. To simulate the ZOH, the control is sampled at the current time (t_0) and then held constant until the next DIDO update is available.

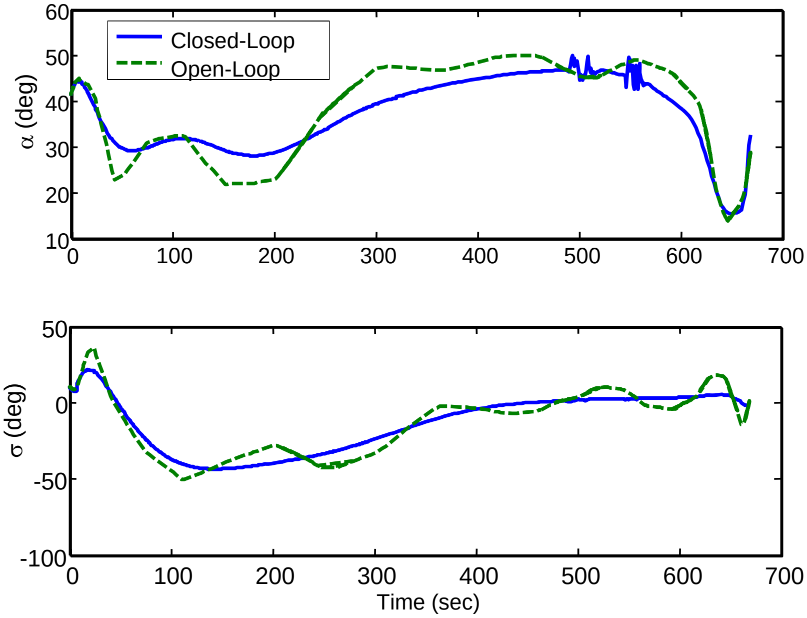


Figure 4.66 Successful Cock-Based PS-Feedback Controls.

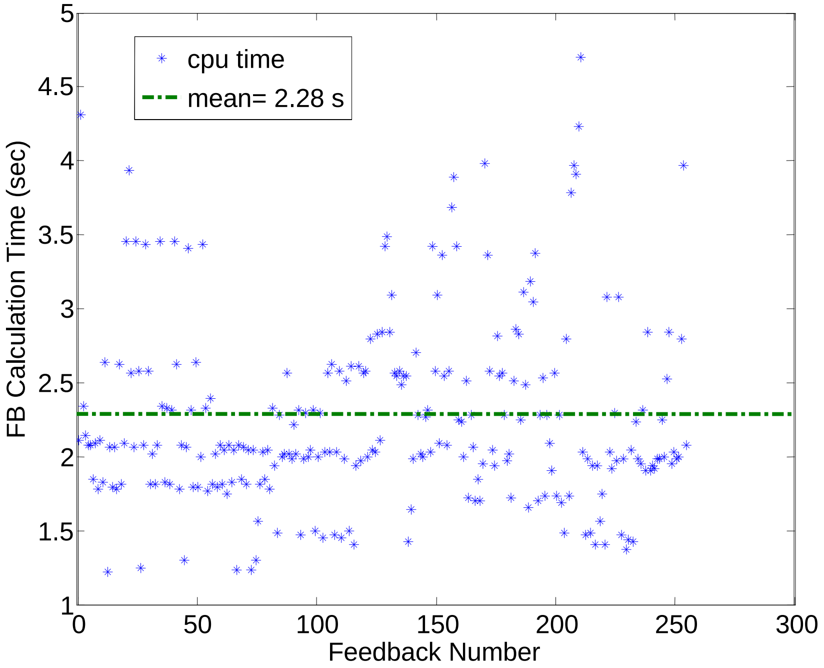


Figure 4.67 Clock-Based Feedback Computation Times.

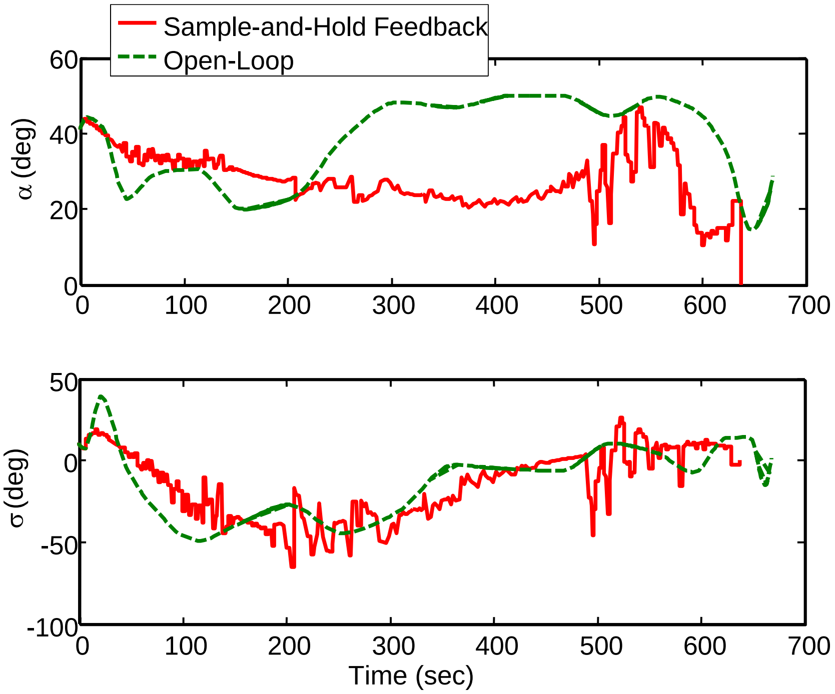


Figure 4.68 Failed Sample-and-Hold PS-Feedback Controls.

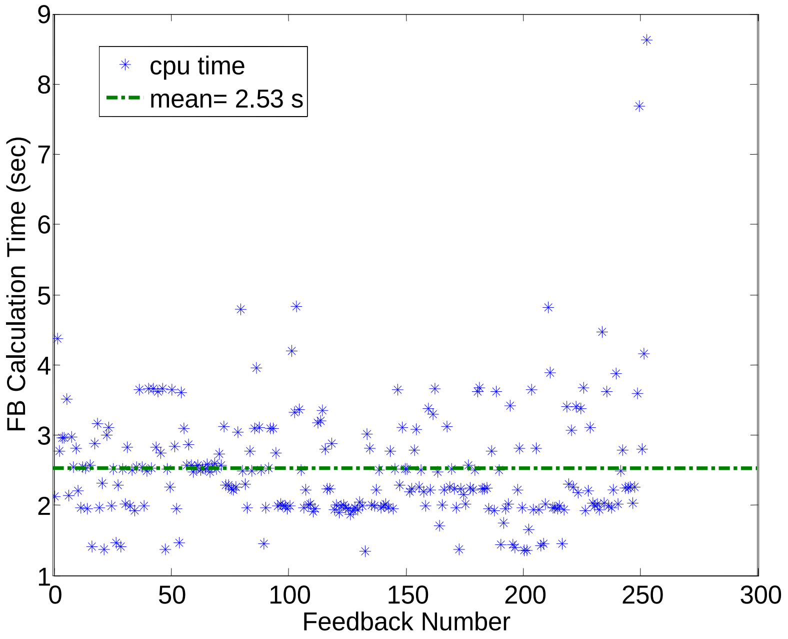


Figure 4.69 Feedback Computation Times for Failed Sample-and-Hold.

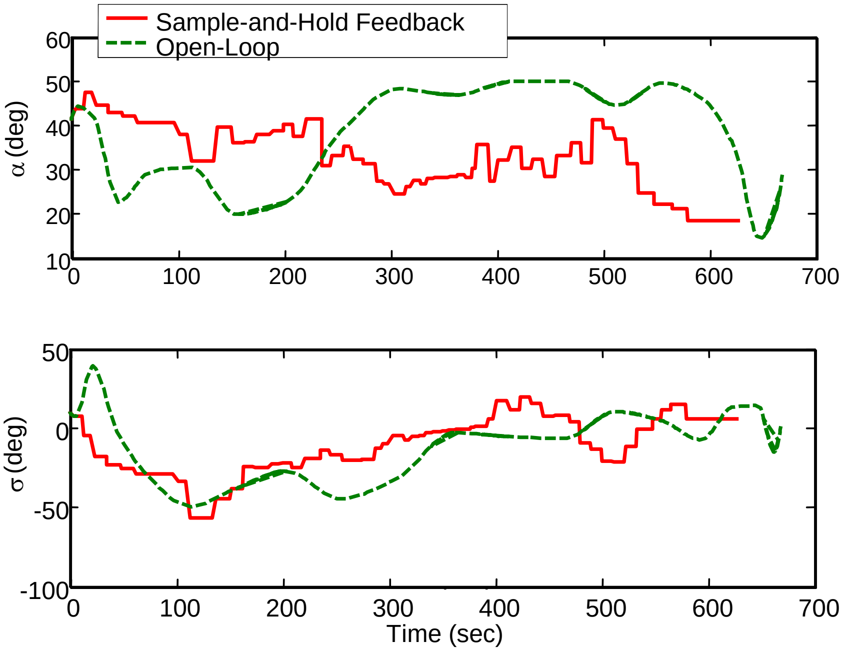


Figure 4.70 Failed Sample-and-Hold PS-Feedback Controls with Added Artificial 8-second Time Delay.

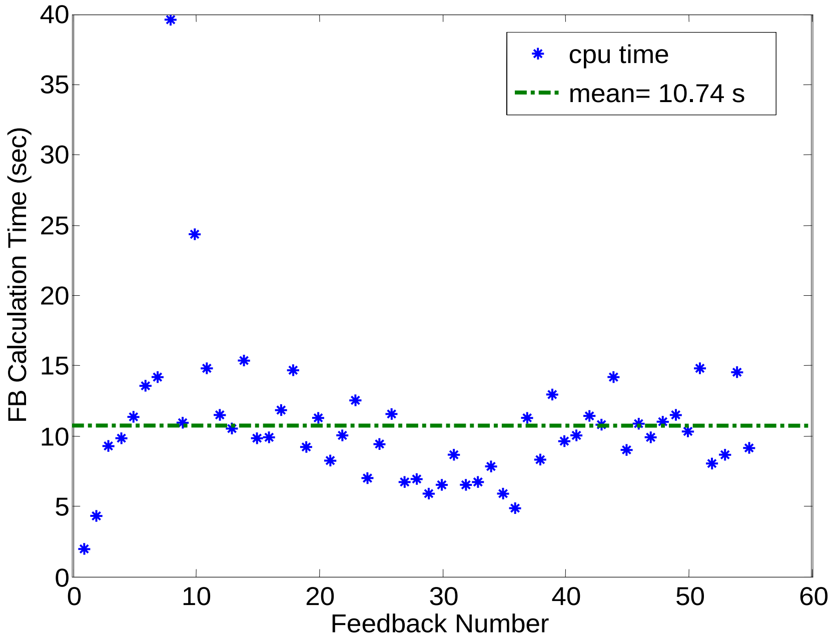


Figure 4.71 Feedback Computation Times for Failed Sample-and-Hold with Added Artificial 8-second Time Delay.

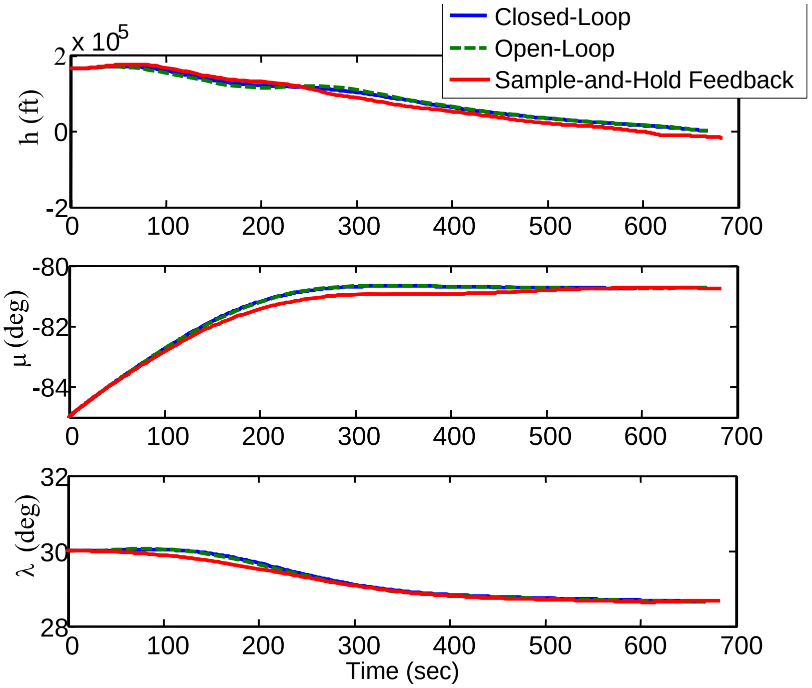


Figure 4.72 Closed-Loop States using Clock-Based and Sample-and-Hold PS-Feedback Controls.

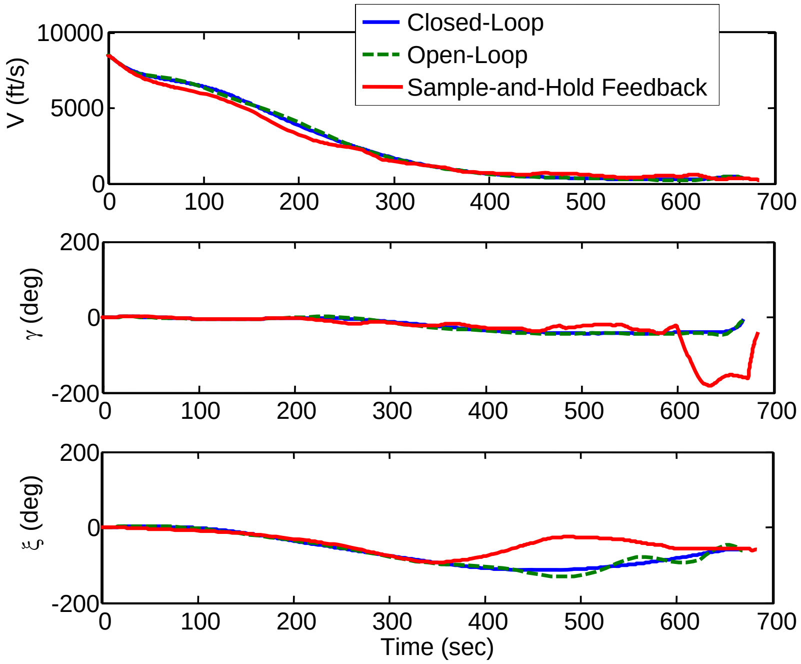


Figure 4.73 Closed-Loop States using Clock-Based and Sample-and-Hold PS-Feedback Controls with Added Artificial 8-Second Time Delay.

Table 4.10 Comparison of Final States for Closed-Loop Solutions

$\mathbf{x}(t_f)$	Desired	CL	Diff.	ZOH	Diff.	ZOH + 8s	Diff.	ZOH @ 1s	Diff.
h (ft)	2000 +/- 400	2093.0879	93.08795	2101.6094	101.60942	-16420.15	18420.154	2089.1243	89.12430
μ (deg)	-80.7112 +/- 0.0011	-80.7112	0.00003	-80.7106	0.00056	-80.74341	0.032	-80.7113	0.00006
λ (deg)	28.6439 +/- 0.0014	28.6438	0.00007	28.6435	0.00042	28.68728	0.043	28.6439	0.00004
V (ft/s)	300 +/- 1.0	299.3251	0.67490	305.2249	5.22492	233.1085	66.892	298.3039	1.69614
γ (deg)	-6 +/- 0.02	-6.0167	0.01671	-7.1076	1.10758	-39.6843	33.684	-5.8848	0.11523
ξ (deg)	-60 +/- 0.02	-59.9879	0.01214	-56.8464	3.15363	-59.55961	0.440	-60.0399	0.03989

The closed-loop control trajectories generated by the method described in Chapter II (i.e., Carathéodory-π solutions) are shown in Figure 4.66 along with the open-loop controls. Having interacted with the plant (i.e., RK4 propagation), the closed-loop control trajectories are indeed different from the open-loop controls. Comparing the clock-based PS-feedback controls with a sample-and-hold (SaH) PS-feedback implementation as shown in Figure 4.66 and Figure 4.68, respectively, reveals that the SaH method fails. Even with comparable computation times, SaH does not converge to the correct solution; hence, verifying the effectiveness of the Carathéodory-π approach as well as validating the Lemma presented in Ref. [148]. The states resulting from both methods are compared in Figure 4.72. Although they appear to be close, some of the states for SaH fail to converge to the required endpoint conditions as depicted in Table

4.10. In addition, Figure 4.73 demonstrates the effects of adding an artificial time delay of 8 seconds making the mean computation time 10.74 sec. By holding the clock and increasing the sampling interval, the required "real-time" for this system is determined to be approximately 10 sec. On the other hand, by artificially reducing the computational time for SaH, the accuracy approaches that of the Carathéodory-π solution for a 1-second update delay; therefore, verifying that SaH requires more than two-times less computational time delay to match the accuracy of Carathéodory-π. Or perhaps more importantly, this implies that a 10-times slower computer is sufficient for the new approach.

Figure 4.74 and Figure 4.75 illustrates wind-gust effects on the vehicle's trajectory for both open-loop and closed-loop simulations, respectively. As seen, the clock-based PS-feedback method corrects for all the wind gusts and safely guides the vehicle to the center of the FAC.

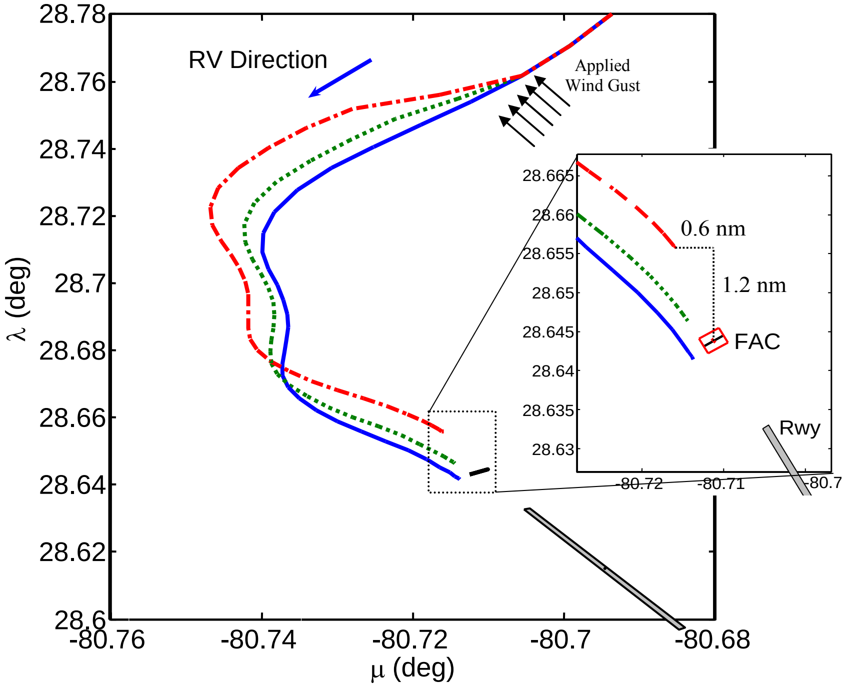


Figure 4.74 Open-loop Ground Tracks with Increasing Wind-Gust Disturbances.

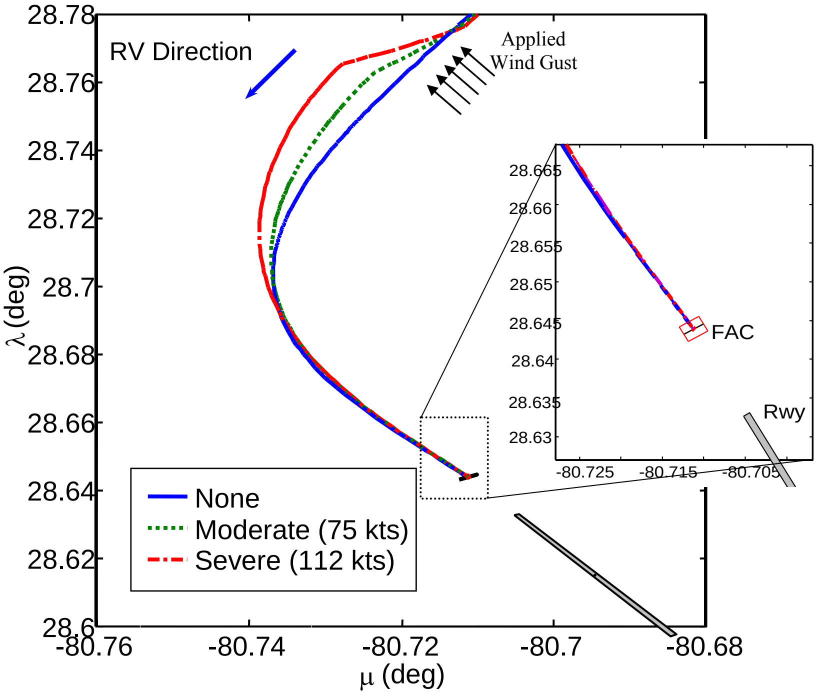


Figure 4.75 Closed-loop Ground Tracks with Increasing Wind-Gust Disturbances.

5. Numerical Considerations

a. Computational Speed

Although the footprint determination in this chapter was simulated manually one range optimization case at a time, it is obvious that onboard applications require very fast computational speed in order to generate a footprint for use in an actual trajectory re-targeting scenario. Since a low-node (~ 20 nodes), bootstrapped solution takes an average of approximately 2 sec for each open-loop solution, then automating the footprint generation would take approximately 12 sec (2 sec x 6 problems). Again, since this work was ran with the Windows and MATLAB overhead and did not involve any modifications to the PS method or the optimization solver, nor emphasize efficient coding techniques to improve speed, it is conceivable that 12 sec could be reduced to 0.12 sec; hence, real-time footprint generation.

In order to improve the quality of the solution as well as the computational speed, scaling values were adjusted and readjusted as necessary. The primary goal when scaling was to ensure that states, controls, and time used by the optimization solver were on the same order of magnitude. Although scaling is critical, not too much time was placed on determining the optimal scale factors as long as a solution was generated in a reasonable amount of time and passed the feasibility and optimality tests.

b. Control Discontinuities

In a few cases involving the closed-loop implementation, small numerical errors in the solution manifested as control "jumps." For example, Figure 4.76 illustrates control jumps encountered in previous models that involved the use of pseudo-controls. 0 zooms in on the first 80 seconds of the trajectory. The figures compare the open-loop solution (solid red lines) with the closed-loop solution (black asterisks) and each complete successive open-loop solution for each feedback iteration (blue dotted lines). Note that the closed-loop solution was only plotted up to approximately 80 sec in order to reduce clutter on the plots.

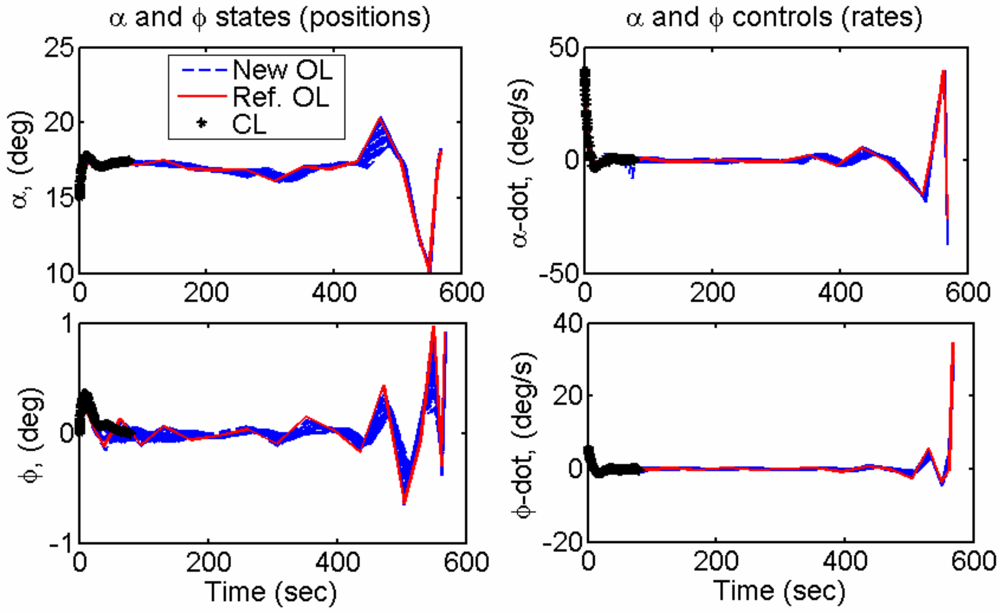


Figure 4.76 Example of Control "Jumps" in a 3-DOF Model using Pseudo-Controls.

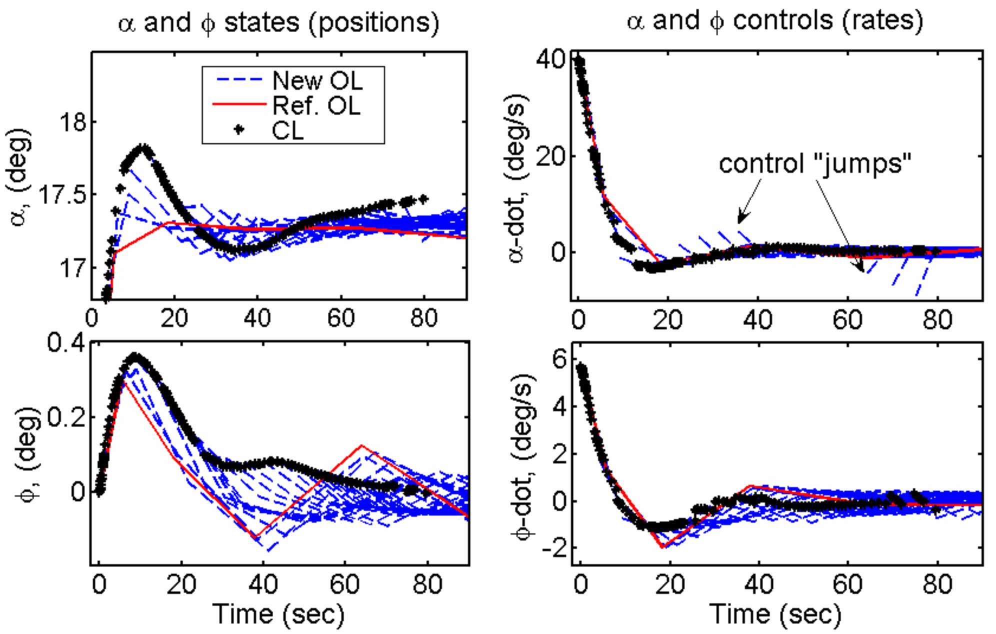


Figure 4.77 A Closer Look at the Control "Jumps."

As demonstrated in 0, more prominently for the AoA-rate given by "α-dot" ($\dot{\alpha} = u_\alpha$), the controls appear to jump. These "jumps" are partly due to the time it takes to compute each open-loop optimal solution since they correspond to the time delay between each feedback update. Unlike the states that must start at the previous position vector, the controls can essentially be reset to a different value within the control constraint bounds. As a consequence of optimal control theory, there is no requirement for the extremal control solution to each successive open-loop optimal control problem to start at the same exact position. Recall that according the Bellman's Principle, if the initial condition for the next run is on the original optimal path, then the solution should be the same; however, the initial condition for each successive run is not exactly on the original optimal path due to numerical errors. As long as these perturbations from the optimal path remain small to within some epsilon-ball then the resulting optimal solution will lie approximately on the original optimal path. This epsilon-ball can be explicitly specified in the problem formulation by adding tolerances on the initial state variables. Similarly, a remedy for the control "jumps" is to place a similar epsilon-ball on the initial control vector for each successive optimization problem, such as $\left|\underline{u}(0)_{k+1} - \underline{u}_k\right| < \varepsilon$, where

ε is some predefined tolerance. Despite the so called control "jumps," as long as they are not too large, then the solution converges to the optimal.

c. Convergence and Infeasible Solutions

A more serious problem is that of convergence. If for some reason, one of the successive open-loop optimization steps does not converge or results in an infeasible solution, it can potentially crash the entire closed-loop process. This can be addressed by designing additional logic into the feedback algorithm to prevent convergence issue from crashing the system. Such logic may include using the previous "good" solution for some finite time and then attempting a new optimization step. In the event of frequent or repetitive non-convergence issues, the only option may be to revert back to using the open-loop response for the remaining trajectory. Of course this would only be practical if the remaining time-to-go is relatively short and there are minimum uncertainties and/or disturbances during this time. If not, at least a "hard" crash is prevented and the possibility of a feasible trajectory still exists. Figure 4.78 shows what an infeasible intermediate open-loop solution looks like where the next solution successfully converges. At about 583 sec into the run, an infeasible solution occurs immediately followed by a feasible solution. Note that the effect is similar to a control jump but to a much larger extent. This can be prevented by flying the previous open-loop solution prior to the infeasibility.

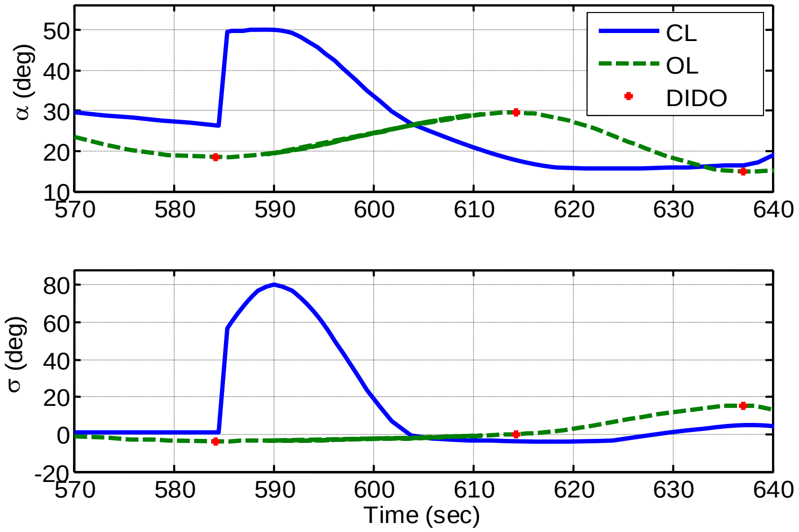


Figure 4.78 Example of an Intermediate Infeasibility in the Feedback Loop.

d. Path Constraint Limits and Infeasibilities

In the course of experimenting with various closed-loop scenarios, a potential problem was discovered involving the implementation of path constraints. For a specific trajectory were the path constraints are active over some finite duration of the flight, it is possible that large numerical errors or external disturbances will push the vehicle beyond the path limits. For the reentry problem, a primary concern is heating during initial entry phase when the vehicle is traveling at excessive speeds. In most cases, the RLV will ride the boundary of the heating rate constraint until the velocity bleeds off. For example, Figure 4.79 compares the heating rate for two trajectories (with and without heating rate constraint) generated using a similar 3-DOF model as used in the previous closed-loop implementation except that the initial conditions now begin at an altitude of 260,000 ft traveling at 24,000 ft/s (M=27).

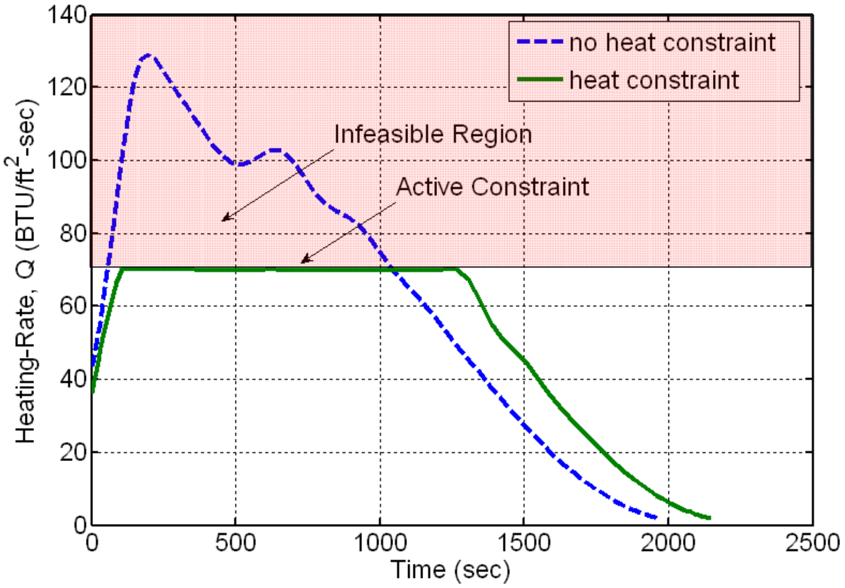


Figure 4.79 Example of Active Path Constraint and Region of Infeasibility.

If during this period of active heat rate constraint an external disturbance is applied that pushes the actual vehicle (simulated by propagation of plant model dynamics) into the infeasible region shown in Figure 4.79, then the current problem formulation used in this closed-loop implementation will fail. In this case, the problem instantaneously goes from feasible to infeasible since the next successive open-loop optimization problem attempts to initiate from an already infeasible position. This is essentially a problem with not having enough margin for the disturbance. Although this can be perceived as more of a practical problem than a numerical problem, one remedy involves imposing a margin of safety by modifying the problem formulation.

One way to make the G&C more robust to uncertainty and disturbance is to implement a penalty function that in effect, allows the computation of a trajectory with more maneuverability margin. The ideal approach would entail a mathematical formulation of a "tube-guidance" problem as shown by the conceptual illustration in Figure 4.80. Here, the goal is to stay in the center of the path constraints defined as a manifold in a higher dimensional Banach space, i.e., hyperspace. Ultimately, the problem is an engineering design problem focused on determining the most effective cost function.

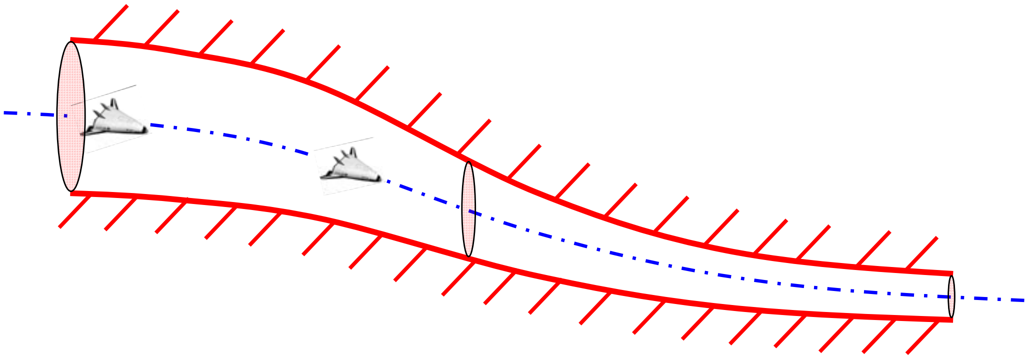


Figure 4.80 Conceptual Idea of "Tube Guidance" Approach.

For example, the objective can be to minimize the maximum heating rate (or other similar path constraint) by using Cheby-Chev optimization, more commonly refered to as a "MiniMax" problem [192]. This secondary objective can be appended to the primary objective to form a weighted, multi-objective (pareto) optimization problem.

D. CONCLUDING REMARKS

This chapter has demonstrated a simple guidance approach, based on optimal control theory, that is capable of generating optimal trajectories with the potential for onboard, autonomous RLV applications. In addition to autonomous, open-loop range maximization, path planning, and footprint generation, the presented feedback approach can provide an RLV capability of optimally satisfying the desired objectives under various disturbances and uncertainties.

The feedback implementation in this chapter helped support recent theoretical developments stipulating that real-time computation of open-loop optimal control implies closed-loop control. In addition, this work verified the requirements of a sampling frequency for generating Carathéodory-π solutions and illustrated the detrimental effects of using a less sophisticated feedback control such as SaH. It was also determined that due to an anti-aliasing effect of this approach, a relatively low degree of discretization (i.e., number of LGL-node points) is sufficient for closed-loop optimal guidance. Thus, it is apparent that this technique is viable for use in optimal guidance algorithms that require corrective maneuvers from the perturbed trajectory.

THIS PAGE INTENTIONALLY LEFT BLANK

V. HIGH-FIDELITY OPTIMAL TRAJECTORY TRACKING

A. INTRODUCTION

Despite efforts involving on-line approaches for reentry guidance and control (G&C), off-line reference trajectories are still used for tracking applications due to the perceived complexity of real-time trajectory planning. As presented in Chapter I, designers have been employing various reference trajectory tracking schemes since the early days of space shuttle entry guidance. Various research has addressed the reentry problem by using an optimal trajectory generator to solve for a reference trajectory, then use other inner-loop control means to track the desired trajectory [52]-[54]. In some cases, off-line reference trajectories are combined with on-line trajectory generators such as the "Optimum-Path-To-Go" methodology developed by Schierman et al. [52].

In a similar fashion, this chapter combines some of the approaches mentioned to demonstrate that a previously developed inner-loop control design, based on dynamic inversion (DI), can successfully track variable body-axis roll, pitch, and yaw commands generated from an off-line, optimal reference trajectory. Note that although the reference trajectory is generated off-line, Chapter IV has shown that real-time trajectory generation is capable of providing optimal nonlinear feedback; hence, making the proceeding work viable for on-line applications.

As an intermediate step towards solving a full, 6-DOF trajectory optimization problem, the overall goal of this chapter is to extend recent developments in the areas of optimal trajectory generation and reconfigurable control by forming a robust G&C architecture that combines three separately developed methods: (1) optimal trajectory generation, (2) guidance command generation based on proportional-integral (PI)-loop closure backstepping, and (3) reconfigurable inner-loop control. The following list identifies the specific objectives for this study.

1. To see how well the inner-loop controller tracks the optimal command histories and remedy any problems
2. To verify what the body-frame angular rates (P,Q,R) should be (steady-state trim values?) since previously assumed constant in other studies

3. To provide initial guesses (if needed) for the 6-DOF optimal reentry trajectory studies using DIDO

4. To provide a baseline for comparing 6-DOF simulation control deflection histories to optimal deflections computed by a 6-DOF DIDO modelTo provide a baseline for future studies involving a single, “integrated” optimal G&C architecture

B. 3-DOF OPTIMAL GUIDANCE, 6-DOF RECONFIGURABLE CONTROL

This section presents the overall G&C design architecture, the specifics about the 3-DOF optimal trajectory generation, command generation, 6-DOF inner-loop reconfigurable tracking controller, and the results related to the maximum downrange and the maximum crossrange optimal control problems.

1. Guidance and Control Design Architecture

The overall G&C design architecture for this work is presented in Figure 5.1. The architecture consists of a two-loop structure: (1) an outer loop that compares the actual angle-of-attack and the bank angle measurements with those provided from the optimal reference trajectory outputs and (2) an inner-loop that is designed to track the optimal body-rates (P,Q,R) generated from the guidance command generator. For this work, a full, 6-DOF model of a RLV was employed for the inner-loop tracking simulation whereas the reference trajectory was generated using a 3-DOF model. Details of the outer and inner loops are presented next.

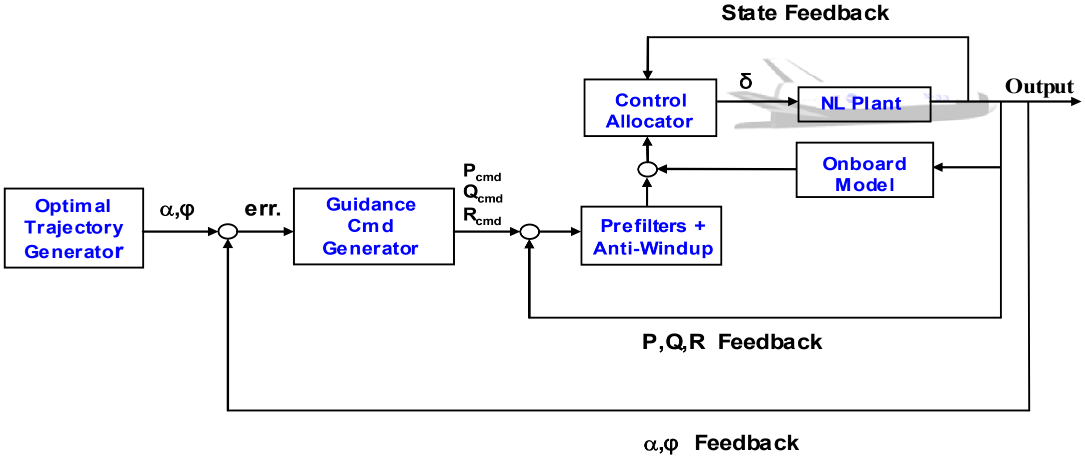


Figure 5.1 Conceptual G&C Design Architecture.

2. Outer-Loop Guidance Command Generation

This section provides some details of how the off-line optimal trajectory is generated with on-line viability and then converted into useful guidance commands.

a. Off-Line, Optimal Trajectory Generation

First, an off-line reference trajectory is generated by posing the reentry problem as a standard optimal control problem (OCP) and solving for the extremal controls using DIDO. A reduced-order model is adequate to demonstrate the feasibility of the approach; therefore, the full 6-DOF equations of motion (EoM) are simplified and decoupled. The outer-loop model assumes a point-mass-model over a flat, non-rotating earth such that the positional and translational EoM in a Cartesian "local horizontal" coordinate system become

$$\begin{aligned}
\dot{x} &= V\cos\gamma\cos\psi \\
\dot{y} &= V\cos\gamma\sin\psi \\
\dot{z} &= V\sin\gamma \\
\dot{V} &= -\frac{D(\alpha,V,z)}{m} - g\sin\gamma \\
\dot{\gamma} &= \frac{L(\alpha,V,z)}{mV} - \frac{g\cos\gamma}{V} \\
\dot{\psi} &= \frac{L(\alpha,V,z)\sin\phi}{mV}
\end{aligned} \tag{5.1}$$

where x (down-range), y (cross-range), and z (altitude) are the vehicle's position with respect to the fixed-earth reference frame, V is the velocity magnitude (i.e., total equivalent airspeed), γ is the flight-path-angle (FPA), ψ is the heading angle (HA), α is the angle-of-attack (AoA), ϕ is the bank angle (BA), m is the vehicle's approximate mass during reentry modeled as 2455 slugs (~79,000 lbs), and g is the standard gravity constant (32.174 ft/s). In Eq. (5.1), the lift and drag forces are represented as L and D, respectively, and are given by

$$L = \frac{1}{2}\rho(z)V^2 C_L(\alpha, M(V,z))S_{ref} \tag{5.2}$$

$$D = \frac{1}{2}\rho(z)V^2 C_D(\alpha, M(V,z))S_{ref} \tag{5.3}$$

where $S_{ref} = 1600\,\text{ft}^2$ is the aerodynamic reference area. The lift and drag coefficients are computed using table lookup data that incorporates wing, body, and trim effects; hence, they are more appropriately represented as

$$\begin{aligned} C_L(\alpha, M(V,z)) &= C_{L_0}(\alpha, M) + C_{L_\delta}(\alpha, M, \underline{\delta}^*(\alpha, M)) \\ C_D(\alpha, M(V,z)) &= C_{D_0}(\alpha, M) + C_{D_\delta}(\alpha, M, \underline{\delta}^*(\alpha, M)) \end{aligned} \tag{5.4}$$

where C_{L_0} represents the wing-body lift coefficient, C_{D_0} represents the sum of the wing-body induced and parasitic drag coefficients, and $\underline{\delta}^*(\alpha, M)$ is the control allocation solution that is discussed later. Likewise, the Mach and density are computed using table lookup data based on a standard 1976 atmospheric model. See Chapter III or Ref. [46] for more details on the use of table lookup data for a similar model.

The optimized controls for this problem are essentially the standard AoA and BA modulation, but to help compensate for command delays (i.e., lags) and to add more realism/fidelity to the problem, as explained in Refs. [46] and [47], the rates of these angles are used as "virtual" controls. This has the benefit of allowing rate limits on AoA and BA which prevents unrealistic responses. Therefore, the control vector is defined as

$$\underline{u} = [u_\alpha,\, u_\phi]^T \in \mathbb{R}^2 \tag{5.5}$$

and the state vector is

$$\underline{x} = [x\ y\ z\ V\ \gamma\ \psi\ \alpha\ \phi]^T \in \mathbb{R}^8 \tag{5.6}$$

As with any dynamical optimization problem, the cost function (objective function), governing EoM, path constraints, boundary conditions, and any constraints (on states and/or controls) must be defined. As such, the general OCP for trajectory generation is fully posed in ChapterII, and repeated here for completeness,

$$\min_u J\left(\underline{x}(\cdot),\underline{u}(\cdot),\tau_0,\tau_f\right)=E\left(\underline{x}\left(\tau_0\right),\underline{x}\left(\tau_f\right),\tau_0,\tau_f\right)+\int_{\tau_0}^{\tau_f} F\left(\underline{x}\left(\tau\right),\underline{u}\left(\tau\right),\tau\right)d\tau$$

$$\text{subject to}\begin{cases}\dot{\underline{x}}(\tau)=\underline{f}\left(\underline{x}(\tau),\underline{u}(\tau),\tau\right)\\ \underline{h}^L\le\underline{h}\left(\underline{x}(\tau),\underline{u}(\tau),\tau\right)\le\underline{h}^U\\ \underline{e}^L\le\underline{e}\left(\underline{x}\left(\tau_0\right),\underline{x}\left(\tau_f\right),\tau_0,\tau_f\right)\le\underline{e}^U\\ \underline{x}^L\le\underline{x}\left(\tau\right)\le\underline{x}^U\\ \underline{u}^L\le\underline{u}\left(\tau\right)\le\underline{u}^U\end{cases}\tag{5.7}$$

where the notation τ represents the computational time as opposed to the physical time as described in ChapterII. The goal for this problem is to maximize the horizontal downrange $\left(x_f\right)$ or cross-range $\left(y_f\right)$ under various constraints; hence, the cost function is

$$Min\,\{\,J[\cdot]\;=-x_f\}\qquad -\text{or}-\qquad Min\,\{\,J[\cdot]\;=\pm\,y_f\}\tag{5.8}$$

subject to the dynamic constraints given by Eq. (5.1), the initial and final event conditions specified as

$$\begin{aligned}&\underline{e}(\underline{x}(t_0),t_0)=\\&(t_0,x_0,y_0,z_0,V_0,\gamma_0,\psi_0,\alpha_0,\phi_0)=(0,0,0,125000\,\text{ft},8417\frac{\text{ft}}{\text{s}},-1.3^\circ,0^\circ,0^\circ,0^\circ)\end{aligned}\tag{5.9}$$

$$\underline{e}(\underline{x}(t_f),t_f)=(z_f,V_f)=(500\,\text{ft},335\frac{\text{ft}}{\text{s}})\tag{5.10}$$

$$-25\frac{\text{ft}}{\text{s}}\le\dot{z}_f\le 8.33\frac{\text{ft}}{\text{s}}\tag{5.11}$$

where $\dot{z}=V\sin\gamma$ and the state (5.12), path (5.13), and control (5.14) inequality constraints, respectively, specified as

$$\begin{bmatrix}0\\-\infty\\0\\0\\-90\,\text{deg}\\-90\,\text{deg}\\-10\,\text{deg}\\-90\,\text{deg}\end{bmatrix}\le\begin{bmatrix}x\\y\\z\\V\\\gamma\\\psi\\\alpha\\\phi\end{bmatrix}\le\begin{bmatrix}\infty\\\infty\\\infty\\\infty\\90\,\text{deg}\\90\,\text{deg}\\50\,\text{deg}\\90\,\text{deg}\end{bmatrix}\tag{5.12}$$

$$\begin{bmatrix} -2.5\text{g's} \\ 0 \\ 0 \end{bmatrix} \le \begin{bmatrix} n_z(z,V,\alpha) \\ \bar{q}(z,V) \\ Q(z,V) \end{bmatrix} \le \begin{bmatrix} 2.5\text{g's} \\ 600\dfrac{\text{lb}}{\text{ft}^2} \\ 60\dfrac{\text{BTU}}{\text{ft-s}} \end{bmatrix} \tag{5.13}$$

$$-40\frac{\text{deg}}{s} \le u_\alpha, u_\phi \le 40\frac{\text{deg}}{s} \tag{5.14}$$

where the path constraint terms represent the normal acceleration $n_z = L(z,V,\alpha)\cos\alpha + D(z,V,\alpha)\sin\alpha$, the dynamic pressure $\bar{q} = \frac{1}{2}\rho(z)V^2$, and the heating rate $Q = k\sqrt{\rho(r)}\,V^{3.15}$ with constant k based on the vehicle's heat shield properties.

b. On-Line, Optimal Trajectory Generation

Although this work computes the optimal reference trajectory off-line and then extracts the appropriate signals to use in the guidance command generation algorithm, preliminary studies conducted concurrently with this work have indicated that the same model using approximated aerodynamic data can solve the problem approximately 85 % faster than using the table look-up data. For example, the 3-DOF work presented in Chapter IV used a second-order polynomial approximation for lift and drag coefficients and a standard two-parameter exponential atmospheric model that resulted in the successful implementation of a nonlinear PS-feedback method with an on-line, trajectory re-optimization scheme that could generate optimal trajectories 99.75 % faster than the same model using the table look-up data [54]. Further work is required to improve the accuracy of the aerodynamic approximations, but initial results look promising for on-line reentry applications.

c. Command Generation via "Backstepping Architecture" (PI & DI)

From the optimal trajectory, the α and ϕ commands are converted into the body-axis angular velocities (P,Q,R) to provide the desired inner-loop commands. The generation of these commands is based on what is known as a "backstepping" approach whereby the "pseudo-commands" at each loop-closure, using PI-control and DI, drives the next inner-most loop [193] . Common loop closures may consist of an outer-most altitude loop, a FPA loop, and an enclosed inner-most AoA loop.

For this experiment, the 3-DOF DIDO trajectory provides the α and ϕ commands that are then used to generate the body-rate commands (P_{cmd}, Q_{cmd}, R_{cmd}). For example, assuming only longitudinal motion, the appropriate pitch rate command is generated based on the following calculations. Ignoring lateral-directional influences (for now), the wind-axis relation $\alpha = \theta - \gamma$ and the simplified pitch rate $Q = \dot{\theta}$ provide the governing EoM such that

$$\dot{\alpha} = -\dot{\gamma} + Q \tag{5.15}$$

Also, the governing EoM for the FPA is

$$\dot{\gamma} = \frac{L(\alpha, V, z)}{mV} - \frac{g\cos(\theta)}{V} \tag{5.16}$$

Substituting Eq. (5.15) into Eq. (5.16), the resulting pitch-rate command is derived as

$$Q_{cmd} = \dot{\alpha}_{des} + \frac{L(\alpha, V, z)}{mV} - \frac{g\cos(\gamma)}{V} \tag{5.17}$$

To improve α tracking, the desired α dynamics are generated using a proportional feedback controller

$$\dot{\alpha}_{des} = K_{\alpha}(\alpha_{cmd} - \alpha) \tag{5.18}$$

where α_{cmd} is the optimal α command from the 3-DOF DIDO trajectory. Figure 5.2 shows a block diagram that represents the computation of the optimal guidance commands.

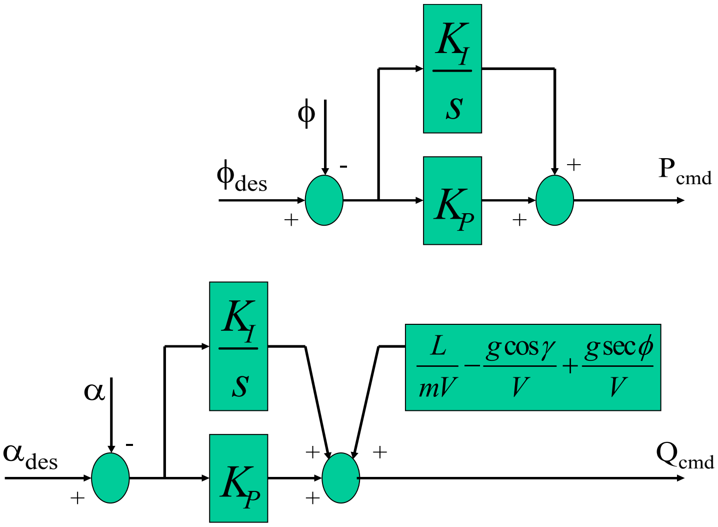


Figure 5.2 P and Q Command Generation.

Note that for the Q_{cmd} generation in Figure 5.2, an extra lateral term is added to account for lateral effects as explained in the results section. Also, not shown in Figure 5.2 is the generation of the yaw command (R_{cmd}) that is computed according to Ref. [168] assuming coordinated turns and is given by

$$R_{cmd} = P_{cmd} \tan\alpha + \frac{g\sin\phi}{u} \tag{5.19}$$

It is also important to note that throughout this section, the notation ϕ is used for both BA and roll angle since they are assumed equal for this preliminary study.

3. AFRL Reconfigurable Inner-Loop Control

This work implemented AFRL's 6-DOF simulation containing a reconfigurable inner-loop control algorithm that uses DI, control allocation, and model following prefilters with integrator anti-windup and reference model bandwidth attenuation. Note that although not repeated here for purpose of brevity, the simulation plant model employs Etkin's standard 6-DOF EoM as defined in Chapter III.

a. Dynamic Inversion and Control Allocation

The inner-loop control system uses DI in order to track the desired body-frame angular velocities $(p_{des}, q_{des}, r_{des})$. The rotational dynamics for this type of vehicle can be written as

$$I\dot{\omega} = G_B - \omega \times I\omega \tag{5.20}$$

where I is the moment-of-inertia tensor, $\omega = [p, q, r]^T$, and G_B is a vector consisting of the total moments acting on the vehicle with contributions from the wing-body-propulsion system (BAE) and the control effectors (δ) such that

$$G_B = G_{BAE}(\omega, P) + G_\delta(P, \delta) = \begin{bmatrix} L \\ M \\ N \end{bmatrix}_{BAE} + \begin{bmatrix} L \\ M \\ N \end{bmatrix}_\delta \tag{5.21}$$

where *L*, *M*, and *N* are the rolling, pitching, and yawing moments, respectively; the vector P denotes a measurable or estimable quantity that can influence body rates and can contain variables such as AoA, sideslip, Mach number, and mass properties; and δ is a vector of control surface deflections given by $\delta = \left[\delta_1, \delta_2, ..., \delta_n\right]^T$. To design the DI control law, Eqs. (5.20) and (5.21) are put into a more standard form by defining $f(\omega, P) \triangleq G_{BAE}(\omega, P) - \omega \times I\omega$ such that

$$I\dot{\omega} = f(\omega, P) + G_\delta(P, \delta) \tag{5.22}$$

The objective is to find a control law that provides direct control over $\dot{\omega}$ such that $\dot{\omega} = \dot{\omega}_{des}$; therefore, the DI control law must satisfy

$$I\dot{\omega}_{des} - f(\omega, P) = G_\delta(P, \delta) \tag{5.23}$$

But, since this problem has more control effectors than control variables, a control allocation algorithm is required to obtain a unique solution.

This control system employs a linear programming-based control allocator which obeys rate and position limits. To implement this type of allocator, the control dependent portion of Eq. (5.23) must be linear in the controls. Hence, Eq. (5.23) is rewritten as

$$I\,\dot{\omega}_{des} - f(\omega, P) = G_\delta(P, \delta) = \tilde{G}_\delta(P)\delta \tag{5.24}$$

In order to account for nonlinearities in the moment-deflection relationship, an intercept term is added to Eq. (5.24) such that

$$I\,\dot{\omega}_{des} - f(\omega, P) = \tilde{G}_\delta(P)\delta + \varepsilon(P, \delta) \tag{5.25}$$

Then, the final inverse control law becomes

$$\dot{\omega}_{des} - f(\omega, P) - I^{-1}\varepsilon(P, \delta) = I^{-1}\tilde{G}_\delta(P)\delta \tag{5.26}$$

For more details on this DI method and the control allocation algorithm see Refs. [39]-[42] and references therein. A block diagram representation of the dynamic inversion control law is shown in Figure 5.3.

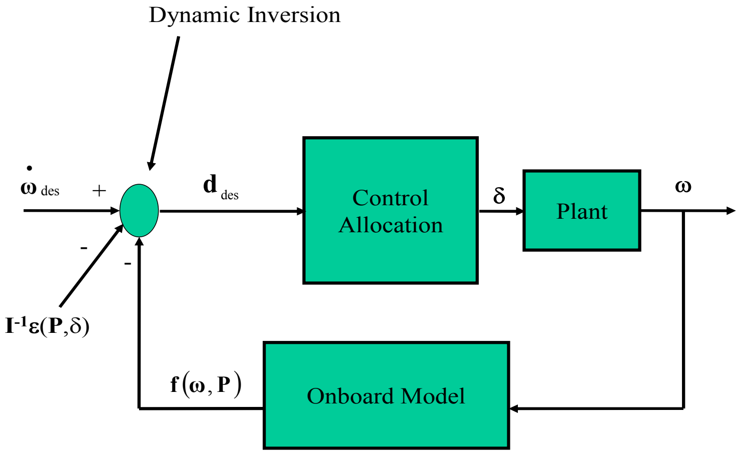


Figure 5.3 Dynamic Inversion with Control Allocation

b. Model Following Prefilters

To provide robustness to modeling errors, inversion errors, and to help shape the closed-loop response, prefilters were added to the DI control system as shown in Figure 5.4. Previous work involving the inner-loop control designs for the X-40A tested two different prefilter structures: implicit and explicit [40]. For this work, an implicit model-following scheme was selected based on its simplicity in regards to having fewer gains that would ultimately need tuning. Also, it was desired that the closed-inner-loop control system from ω_{des} to ω has the characteristics of a first-order response. The implicit structure presented in Figure 5.5 provides this behavior and helps compensate for imperfections in the DI control law. A closer look at this structure with some straight-forward block diagram algebra reveals that a stable pole/zero cancellation occurs. The resulting transfer function displays the desired closed-inner-loop response

$$\frac{\omega}{\omega_{des}} = \frac{\frac{K_b}{2}}{s + \frac{K_b}{2}} \tag{5.27}$$

Note that Figure 5.5 only displays a single loop; however, the actual model implemented contained a loop for each of the body-axis angular rates.

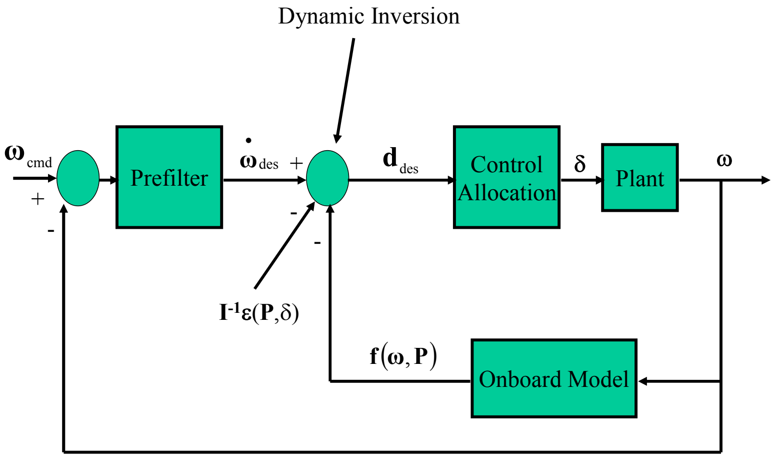


Figure 5.4 Dynamic Inversion with Control Allocation and Prefilters.

c. *Integrator Anti-Windup and Reference Model Bandwidth Attenuation*

Axis saturation occurs when all control power is used on one or more axes. For flight control applications, when a control surface moves at its rate limit or resides on a position limit, then control effector saturation occurs. This is a necessary, but not sufficient, situation for axis saturation. With axis saturation, no additional control power is available when requested by the control system and this should be taken into account by the control law. Analysis of the control allocation inputs ($\mathbf{d}_{\text{des}}$) and outputs ($\mathbf{B\delta}$) can indicate axis saturation. To prevent attempts to cancel tracking errors caused by axis saturation, the following integrator anti-windup law is added to reduce the magnitude of input signal to the integrator when axis saturation occurs.

$$I_{AW} = K_{AW}(\mathbf{B\delta}_{cmd} - \mathbf{d}_{des}) \tag{5.28}$$

where K_{AW} is the anti-windup gain, $\mathbf{d}_{des}$ are the desired accelerations from the control effectors, and $\mathbf{B\delta}_{cmd}$ is the acceleration that the control allocator effectiveness model predicts is being produced by the effectors. If no saturation occurs, then $\mathbf{B\delta}_{cmd} - \mathbf{d}_{des} = 0$ and the control law operates normally; otherwise, at least one axis is saturated and the state of the prefilter integrator is reduced by the anti-windup signal. The anti-windup scheme is implemented as depicted in Figure 5.5.

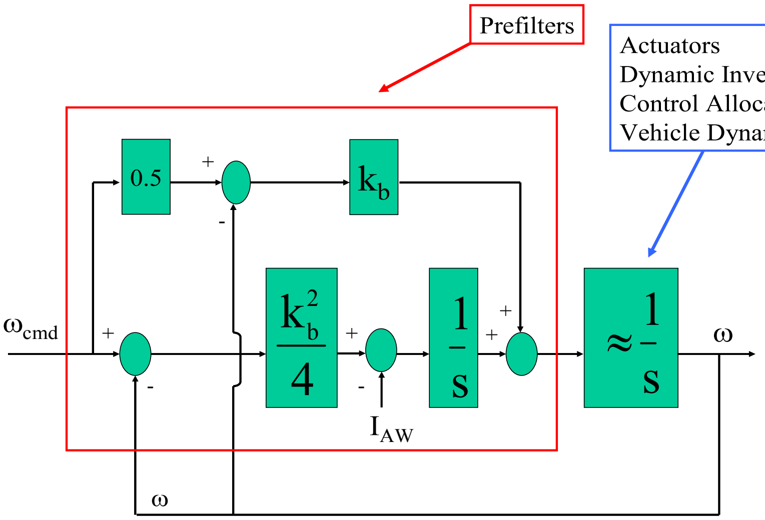


Figure 5.5 Implicit Model Following Prefilter Integrator Anti-Windup Compensation.

For more details on the anti-windup integrator and its use see Ref. [39] and references therein.

4. Results and Discussions

The primary performance concerns for this work were tracking error and control saturation. Of course the overall index of performance is that the cost functions for both the outer-loop guidance and inner-loop control agree to within an acceptable tolerance. As long as the desired trajectory and cost were accomplished, the tracking performance was only graphically confirmed. For this work, only the max downrange and max cross-range results are presented.

a. Max Downrange (DR) Case

For the case of maximizing the vehicle's downrange distance, the 3-DOF optimal trajectory is generated off-line and α and ϕ histories are extracted for use as the desired guidance commands. These command profiles are shown in Figure 5.6.

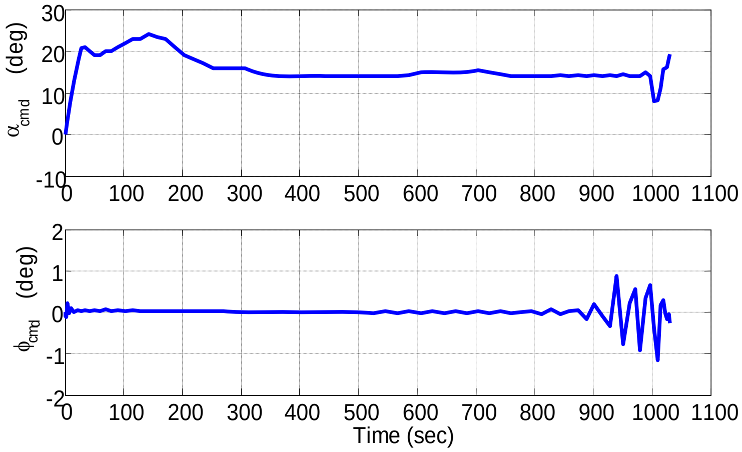


Figure 5.6 Max DR Commands from 3-DOF Optimal Reference Trajectory.

As with most complex control system design, there were some initial problems that had to be resolved before the inner-loop controller could successfully track the guidance commands. The less trivial problems had to do with initial transients and steady-state errors in the simulation.

The first few attempts of running the simulation revealed problems with initial transients that took the 6-DOF controller too long to recover from, or in some cases, no recovery at all. It was quickly determined that poor initial conditions specified in the 6-DOF simulator caused immediate control saturation; hence, leading to un-recoverable instabilities. To resolve this, feasible initial trim conditions were determined by a three-step process. First, initial altitude and velocities were verified by picking off data points from a previously published reference trajectory for the same vehicle. Next, the trim condition was approximated by using the standard 6-DOF EoM. These results were then verified using a graphical method based on first principles.

Assuming only longitudinal trim by ignoring later-directional motion, a vehicle gliding at a constant descent will have no roll ($p = 0$) or yaw ($r = 0$) and no normal translation ($v = 0$). Also, the following rates will be zero as well.

$$\dot{u} = \dot{v} = \dot{w} = \dot{p} = \dot{q} = \dot{r} = 0 \tag{5.29}$$

Forces	$X = m(\dot{u} + qw - rv + g\sin\theta)$ $Y = m(\dot{v} + ru - pw - g\cos\theta\sin\phi)$ $Z = m(\dot{w} + pv - qu - g\cos\theta\cos\phi)$
Moments	$L = I_x\dot{p} - I_{xz}(\dot{r} + pq) + (I_z - I_y)qr$ $M = I_y\dot{q} + I_{xz}(p^2 - r^2) + (I_x - I_z)rp$ $N = I_z\dot{r} - I_{xz}(\dot{p} - qr) + (I_y - I_x)pq$
Body Angular Velocities	$p = \dot{\phi} - \dot{\psi}\sin\theta$ $q = \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi$ $r = -\dot{\theta}\sin\phi + \dot{\psi}\cos\theta\cos\phi$
Euler Rates	$\dot{\phi} = p + \dot{\psi}\sin\theta$ $\dot{\theta} = q\cos\phi - r\sin\phi$ $\dot{\psi} = r(\frac{\cos\phi}{\cos\theta}) + q(\frac{\sin\phi}{\cos\theta})$

Figure 5.7 6-DOF Equations of Motion used for the Simulation's Inner-Loop Controller.

Therefore, the equations in Figure 5.7 reduce to

$$\begin{aligned} &X - mg\sin\theta = 0 \\ &Y = 0 \\ &Z + mg\cos\theta = 0 \\ &L = M = N = 0 \\ &p = q = r = 0 \\ &\dot{\phi} = \dot{\theta} = \dot{\psi} = 0 \end{aligned} \tag{5.30}$$

For there to be zero pitching moment, the base effects (BAE) plus the control surface contributions (delta) must sum to zero,

$$M = 0 \;\Rightarrow\; M_{BAE} + M_{\delta} = 0 \tag{5.31}$$

Now, assuming symmetric control surface deflections,

$$\begin{aligned}
\delta_{RE_{in}} &= \delta_{LE_{in}} &&\Rightarrow & \delta_{RE_{in}} - \delta_{LE_{in}} &= 0 \\
\delta_{RE_{out}} &= \delta_{LE_{out}} &&\Rightarrow & \delta_{RE_{out}} - \delta_{LE_{out}} &= 0 \\
\delta_{RF} &= \delta_{LF} &&\Rightarrow & \delta_{RF} - \delta_{LF} &= 0 \\
\delta_{RR} &= \delta_{LR} &&\Rightarrow & \delta_{RR} - \delta_{LR} &= 0
\end{aligned} \tag{5.32}$$

and the approximate angle relation $\gamma = \theta - \alpha$, the trim conditions for α, δ_{RE}, δ_{LE}, δ_{RF}, δ_{LF}, δ_{RR}, δ_{LR}, and θ are solved using MATLAB's "fsolve" function. The results for at $h = 125{,}000\,\text{ft}$ are given in Table 5.1.

Table 5.1 Initial Trim Condition Results from MATLAB's "fsolve"

Angles (deg)	**u = 4146.9 ft/s**	**u = 5118.5 ft/s**
α	19.5798	15.4031
$\delta_{RE_{in}}$	-0.8112	-0.2664
$\delta_{LE_{in}}$	-0.8112	-0.2664
$\delta_{RE_{out}}$	-0.8112	-0.2664
$\delta_{LE_{out}}$	-0.8112	-0.2664
δ_{RF}	7.8910	10.9018
δ_{LF}	7.8910	10.9018
δ_{RR}	0	0
δ_{LR}	0	0
θ	-18.0830	-26.1053
γ	-37.6628	-41.5083

As a sanity check, the trimmed flap settings can be graphically confirmed by plotting the total pitching moment coefficients verses the total flap deflection for various angles of attack. For a given angle-of-attack, the trimmed flap deflection corresponds to the point where the curve crosses the zero pitching moment. As shown in Figure 5.8 and Figure 5.9, the trimmed flap settings correspond to those approximated by "fsolve" as indicated by the highlighted cells in Table 5.1. Using these trim values as the initial conditions in the simulation removed the transients and provided smooth system response.

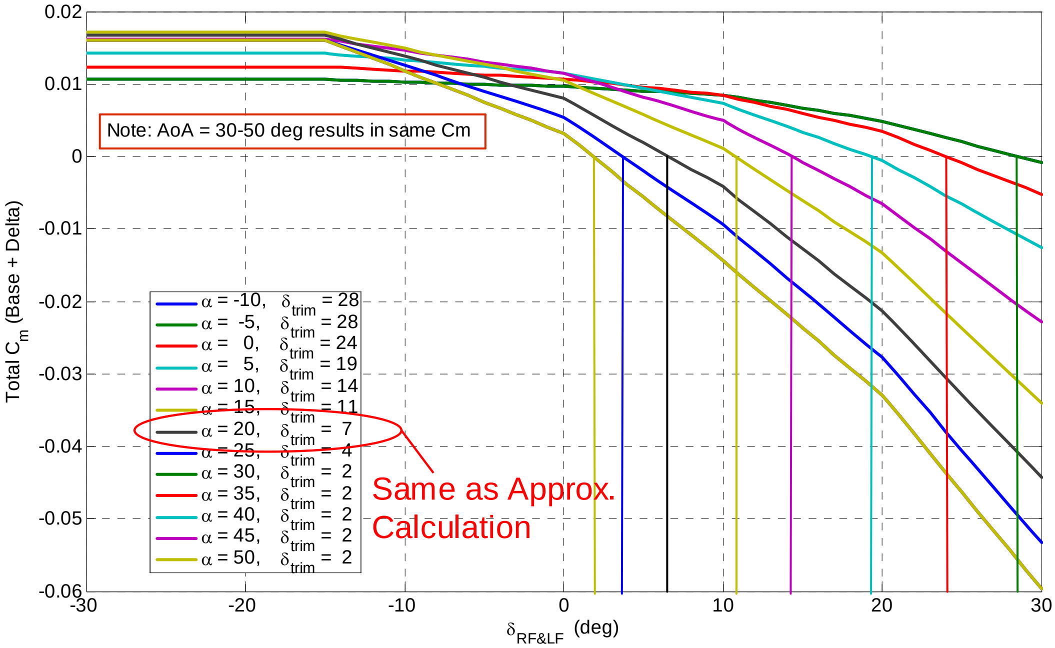


Figure 5.8 Total Pitching Moment vs. Flap Defelction @ Mach 4.6 for Various Angles of Attack.

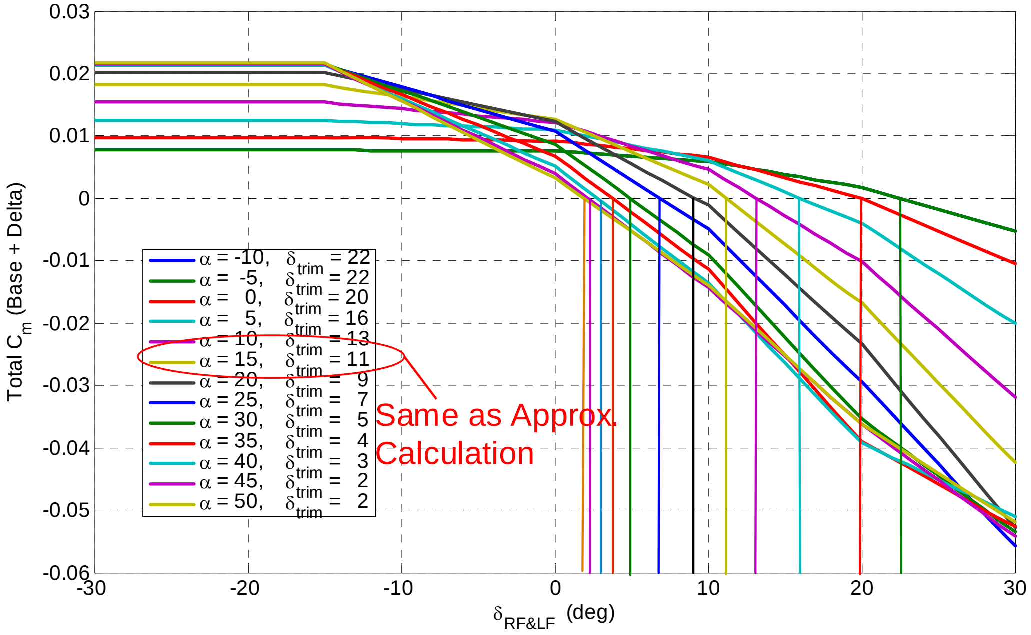


Figure 5.9 Total Pitching Moment vs. Flap Deflection @ Mach 8.0 for Various Angles of Attack.

With the trim issue resolved, initial comparison of the PQR-guidance commands with the actual PQR-states revealed that the inner-loop controller was successfully tracking the desired values; however, when comparing the actual states to those of the reference trajectory, there were some unacceptable errors, especially for α. After carefully reviewing the data, trial-and-error gain tuning on the prefilter and anti-windup gains, it was determined that adding an integrator in the command generation block (see Figure 5.10), improves the reference trajectory tracking as shown in Figure 5.11.

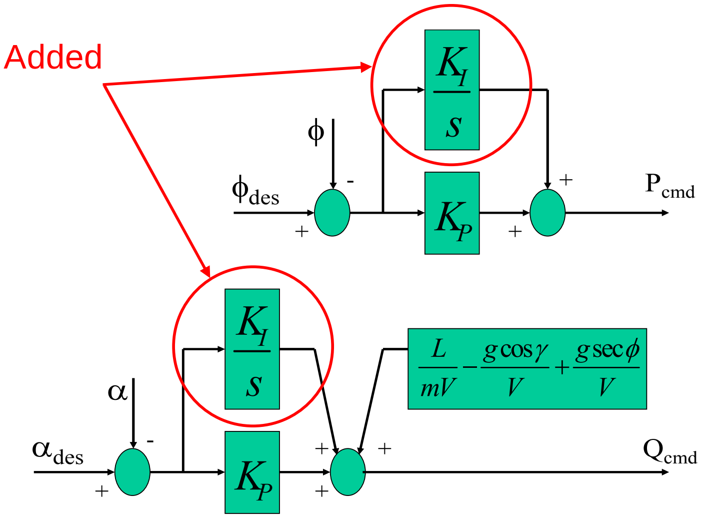


Figure 5.10 Modification of P and Q Command Generation.

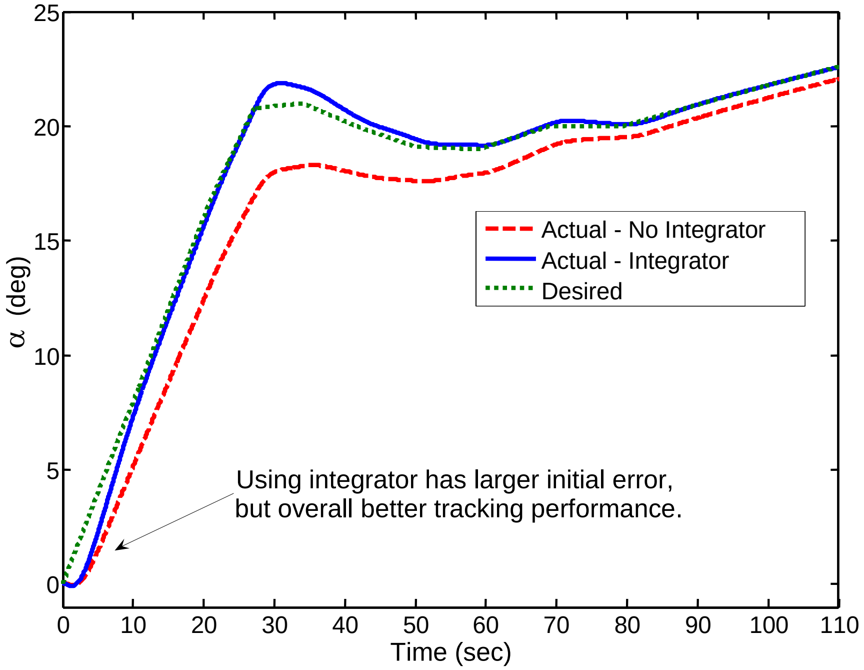


Figure 5.11 Comparison of AoA with and without Integrator in Command Generation Logic.

With this addition and use of the gains in Table 5.2, the max DR results were acceptable with an average difference of only 2 % between the actual and commanded/desired values as seen in Figure 5.14 to Figure 5.17. The cost for the optimal reference trajectory and the simulation were 1,515,588 ft and 1,515,852 ft, respectively, which results in only a 0.017 % error.

Table 5.2 Tuned Gains used in Inner-Loop Control System

Gain Type	Max DR	Max CR
Prefilter BW, P (K_{bP})	5.0	4.0
Prefilter BW, Q (K_{bQ})	5.0	4.0
Prefilter BW, R (K_{bR})	5.0	4.0
Proportional DI (K_P)	0.8	0.9
Integral DI (K_I)	0.5	0.5
Anti-Windup, P ($K_{AW,P}$)	0.2	0.1
Anti-Windup, Q ($K_{AW,Q}$)	0.2	0.1
Anti-Windup, R ($K_{AW,R}$)	0.2	0.1

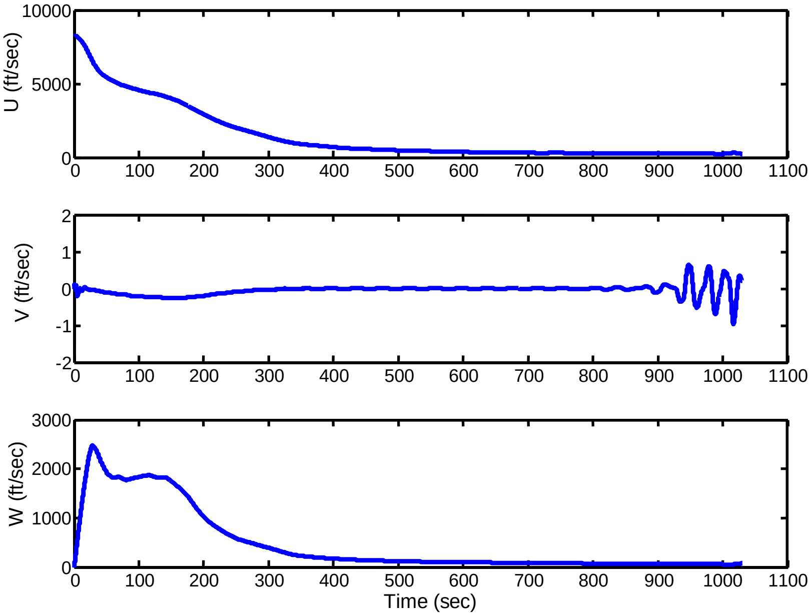


Figure 5.12 Max DR Linear Velocities (U,V,W).

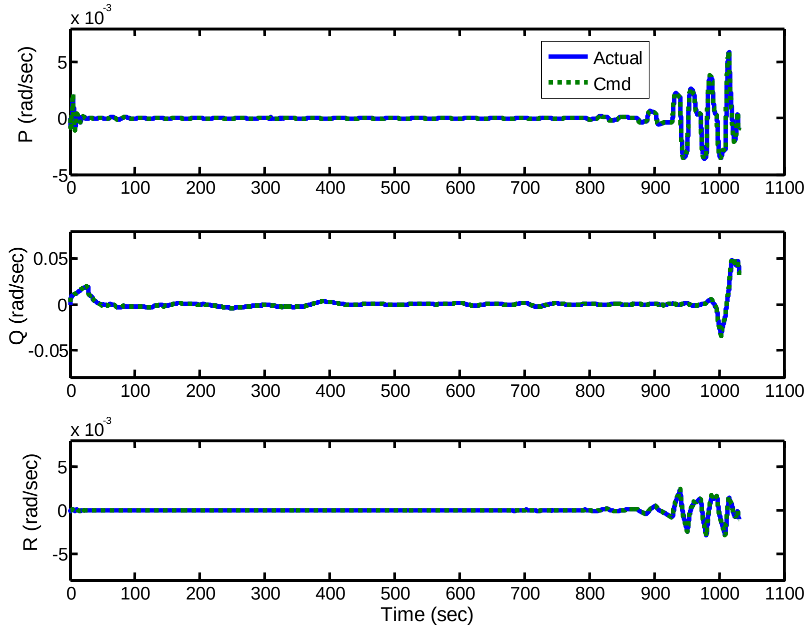


Figure 5.13 Max DR Angular Body-Rates (P,Q,R).

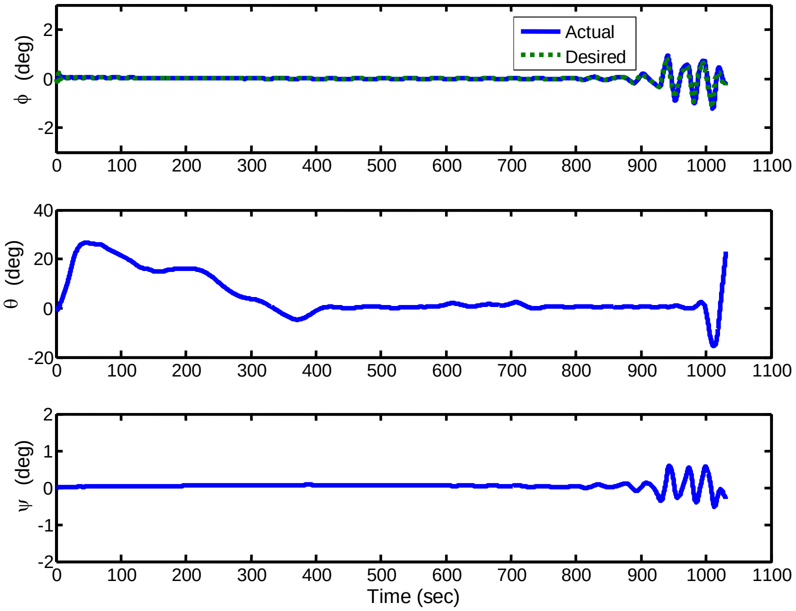


Figure 5.14 Max DR Euler Angles (Phi, Theta, Psi).

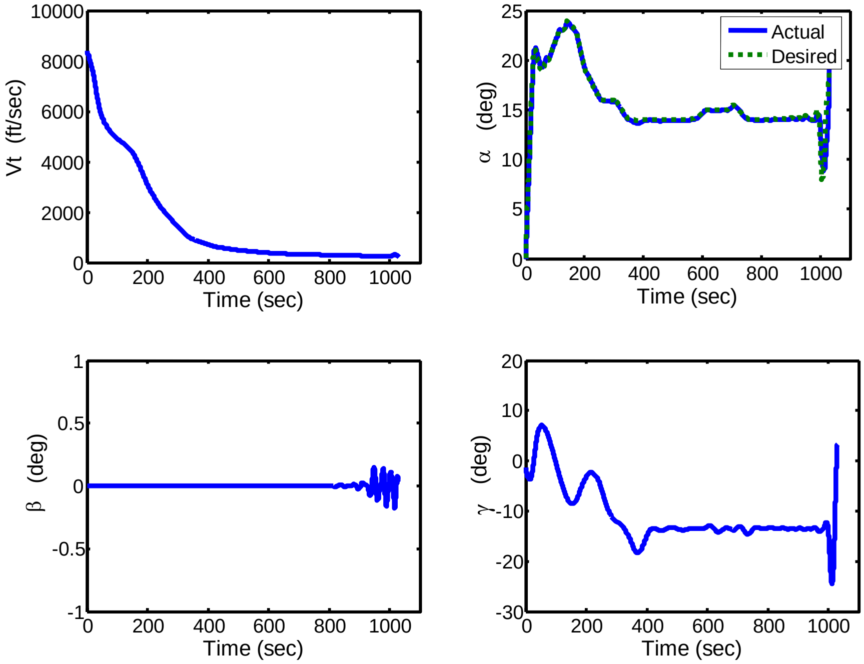


Figure 5.15 Max DR Airspeed (V) and Wind-Relative Angles (Alpha, Beta, Gamma).

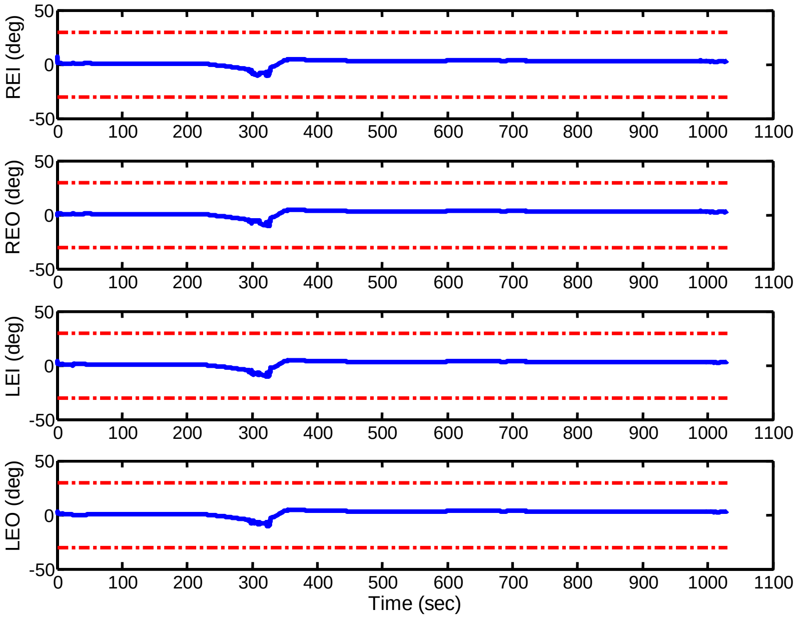


Figure 5.16 Max DR Elevon Control Surface Deflections (Right Elevon Inboard, etc).

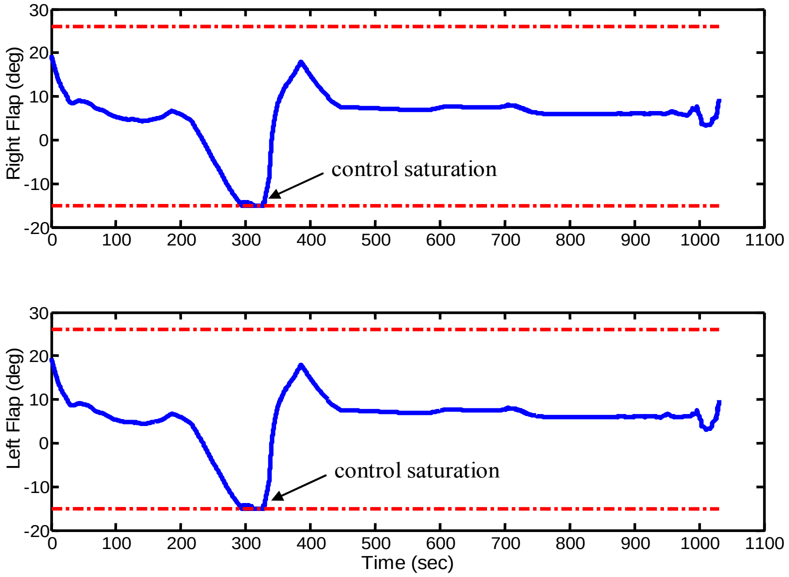


Figure 5.17 Max DR Body Flap Control Surface Deflections.

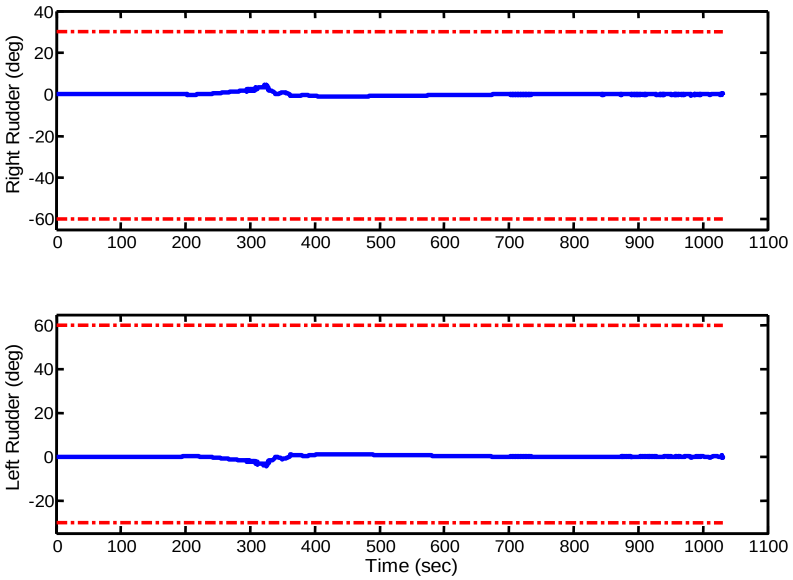


Figure 5.18 Max DR Rudder Control Surface Deflections.

Figure 5.16 to Figure 5.18 show that all of the control surface deflections in the 6-DOF simulation remain within their respective limits as indicated by the dash-dot lines. As indicated in Figure 5.17, there is only one region of body-flap control saturation from approximately 290-325 sec that may be related to an initial pitch down attitude just prior to what appears to be a steady-state trimmed condition from approximately 425-980 sec. Figure 5.19 shows that the anti-windup values remain approximately zero (10^{-15}) throughout the flight simulation.

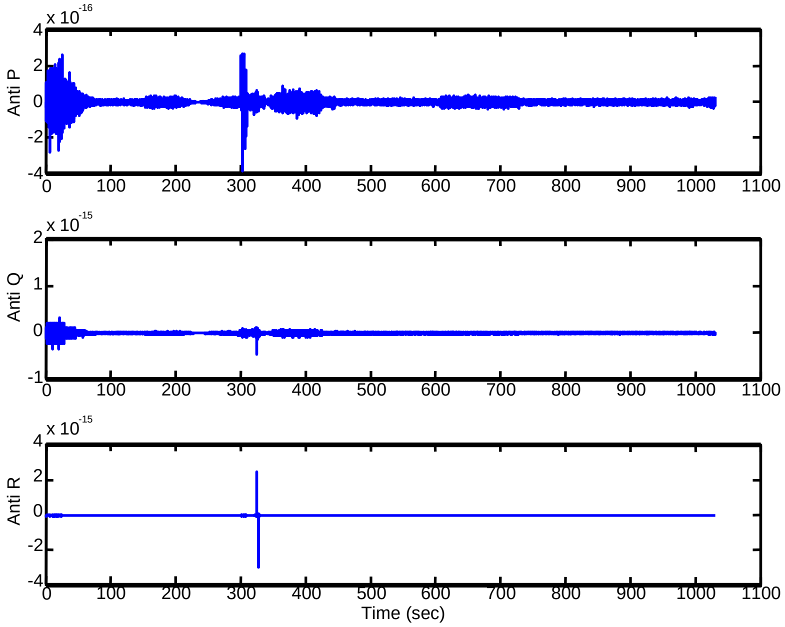


Figure 5.19 Max DR Anti-Windup Signals.

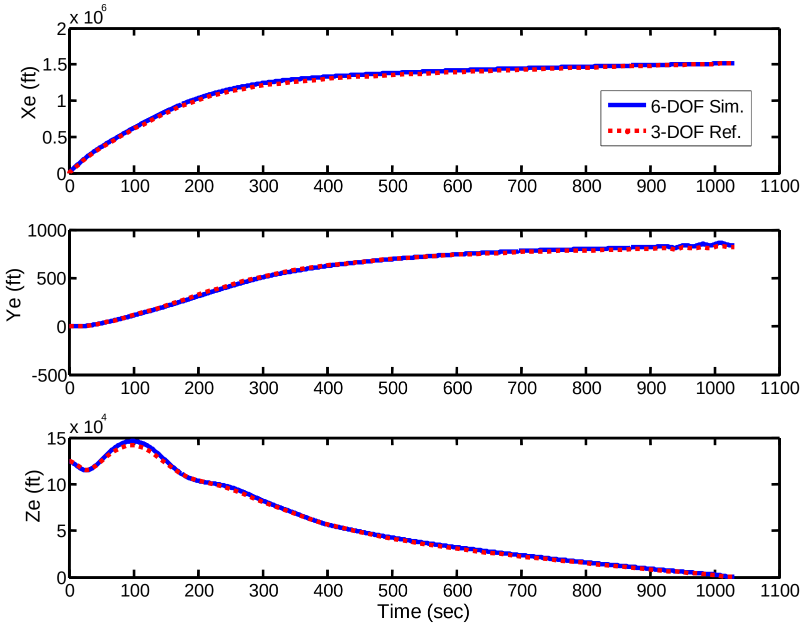


Figure 5.20 Max DR Comparison of Reference and Tracking Trajectory.

In addition to checking that the system adequately tracks the desired guidance commands, Figure 5.20 shows that these commands result in an acceptable trajectory when comparing the 6-DOF simulation and the 3-DOF reference x,y,z-state histories. The corresponding forces and moments are shown in Figure 5.21 to Figure 5.26.

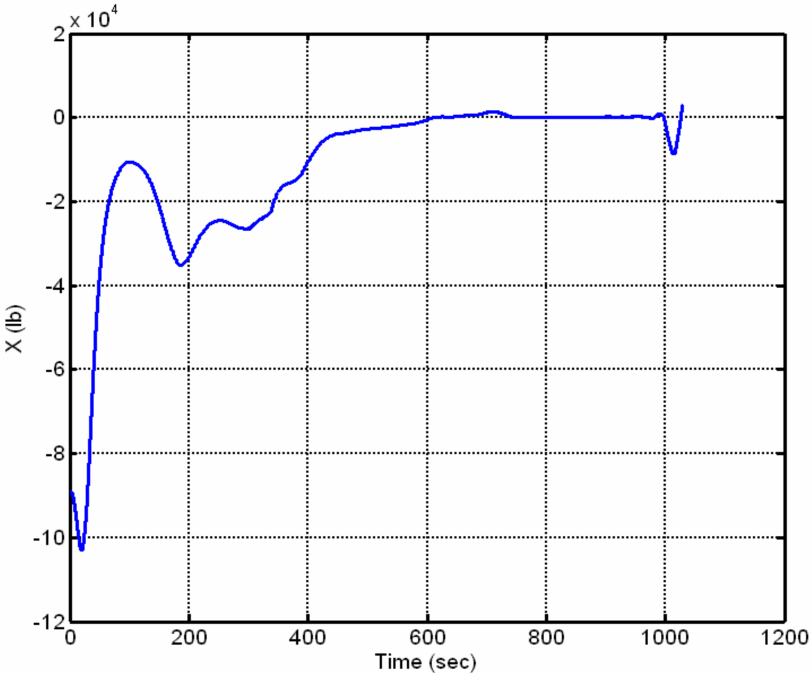


Figure 5.21 Axial Force History for Max DR Trajectory.

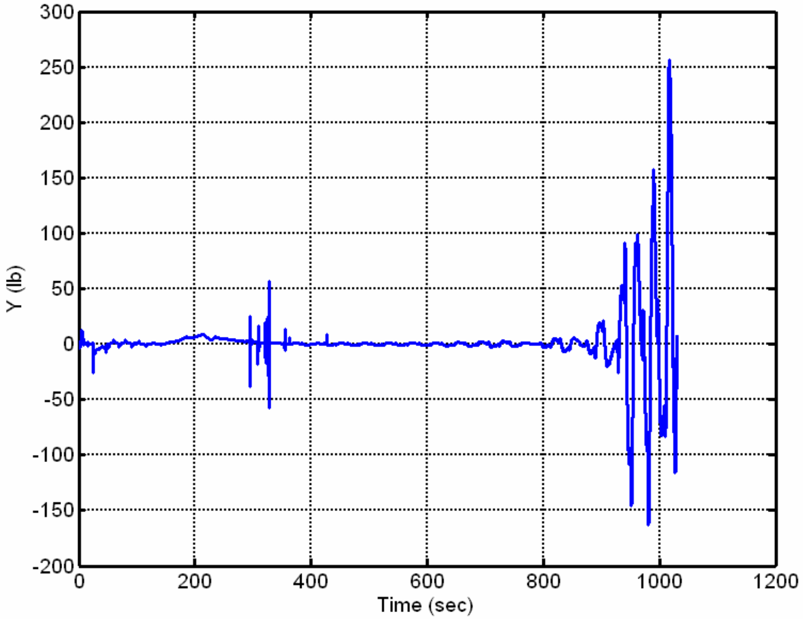


Figure 5.22 Side Force History for Max DR Trajectory.

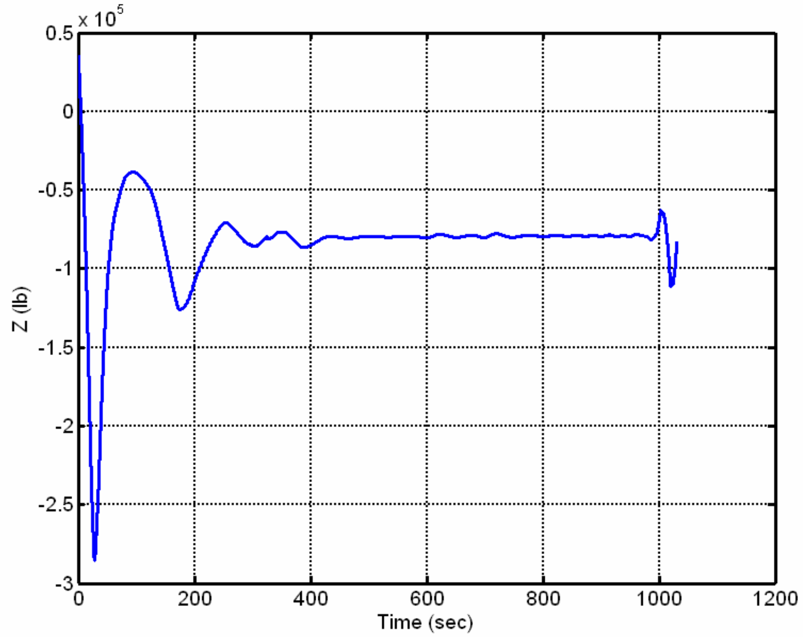


Figure 5.23 Normal Force History for Max DR Trajectory.

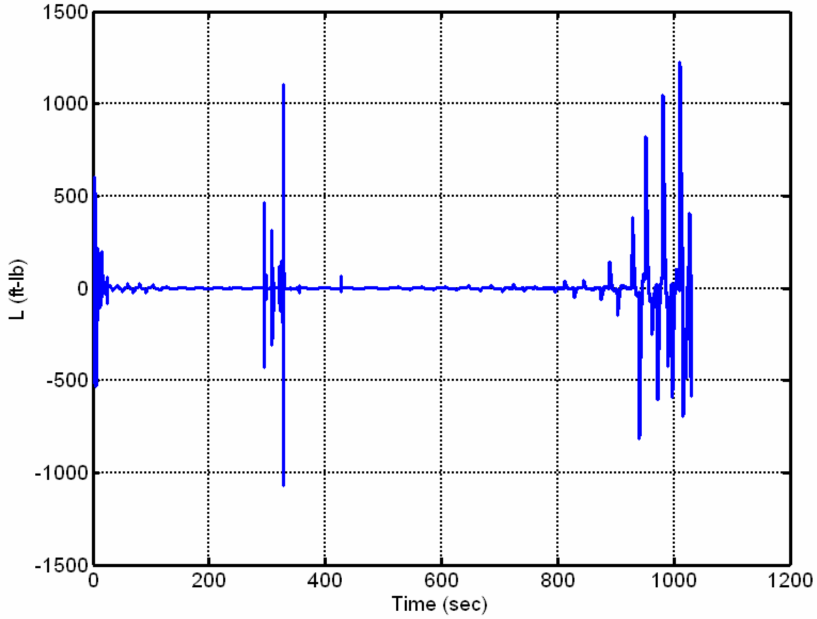


Figure 5.24 Rolling Moment History for Max DR Trajectory.

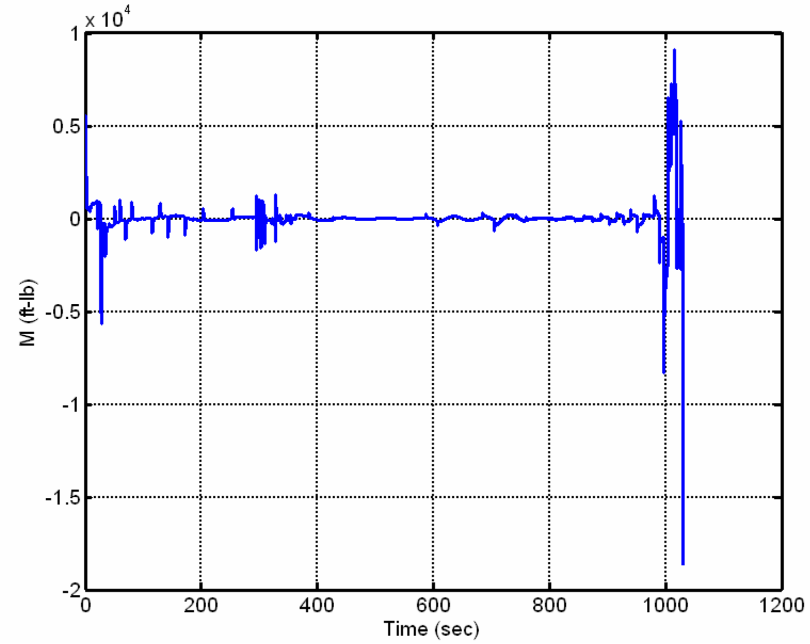


Figure 5.25 Pitching Moment History for Max DR Trajectory.

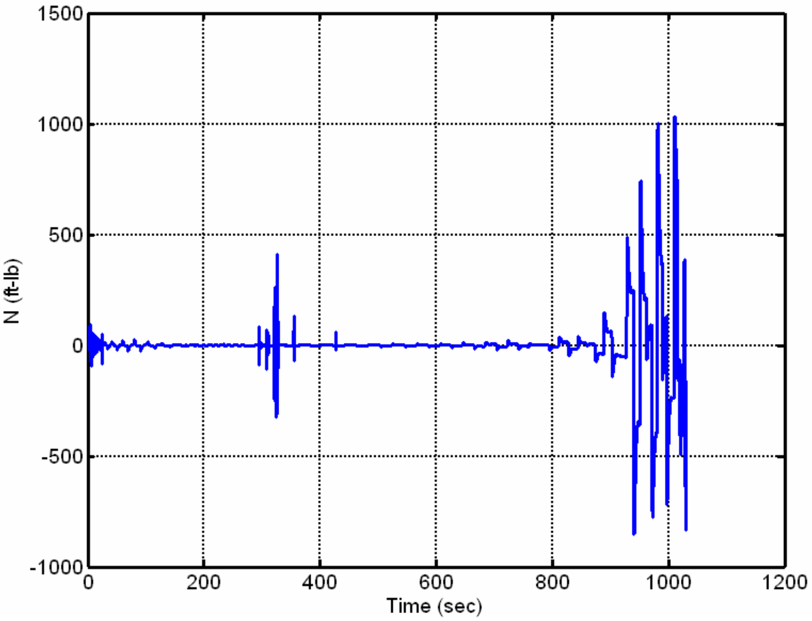


Figure 5.26 Yawing Moment History for Max DR Trajectory.

b. Max Crossrange (CR) Case

For the case of maximizing the vehicle's cross-range distance, α and ϕ histories are extracted from the off-line optimal trajectory as was done for the max DR case. These command profiles are shown in Figure 5.27.

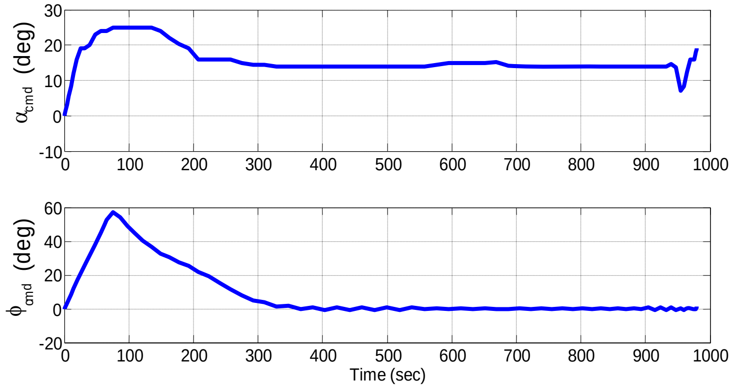
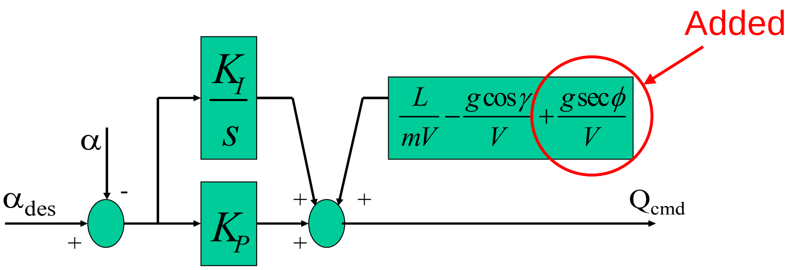


Figure 5.27 Max CR Commands from 3-DOF Optimal Reference Trajectory.

For maneuvers with large bank angles, such as the max CR case, there is a loss of lift that must be countered in order to maintain altitude. Typically, the elevator is used to increase the effective wing-body angle-of-attack; hence, increasing lift. In short, the longitudinal and lateral equations of motion are strongly coupled during maneuvers with large bank angles. For example, a pilot must maintain adequate backpressure on the yoke during steep turns to prevent loss of altitude. To account for this in the pitch-command (Q_{cmd}) generation logic, a secant term was added as shown in Figure 5.28 to provide an approximate contribution from the bank angle according to the following relation [99]

$$L = mg\sec(\phi) \Rightarrow \frac{g}{V}\sec(\phi) \tag{5.33}$$



Added
$\frac{K_I}{S}$
K_P
$\frac{L}{mV} - \frac{g\cos\gamma}{V} + \frac{g\sec\phi}{V}$
α
α_{des}
-
+
+
+
+
Q_{cmd}

Figure 5.28 Modification of P and Q Command Generation.

Similarly, the simplifying assumption that $\dot{\theta} = Q$ used for the backstepping architecture in section III.C is only valid when the roll angle (ϕ) is sufficiently small. For the maximum CR trajectory where the roll angle may be large, the following relation is used $\dot{\theta} = Q\cos\phi - R\sin\phi$ [99] such that Eq. (5.15) becomes

$$\dot{\alpha} = -\dot{\gamma} + Q\cos\phi - R\sin\phi \tag{5.34}$$

and Eq. (5.17) becomes

$$Q_{cmd} = \left[\dot{\alpha}_{des} + \frac{L(\alpha, V, z)}{mV} - \frac{g\cos(\gamma)}{V} + R\sin\phi\right]\sec\phi \tag{5.35}$$

Of course, this still has the implied assumption that $\alpha = \theta - \gamma$ which in only valid for "wings-level" flight, but including Eq. (5.33) helps compensate for this.

With this modification and use of the gains in Table 5.2, the max CR results were acceptable with only an average difference of 2.2 % between the actual and commanded/desired values as seen in Figure 5.29 to Figure 5.32. The cost for the optimal reference trajectory and the simulation were 664,862 ft and 671,781ft, respectively, which results in only a 1.04 % error.

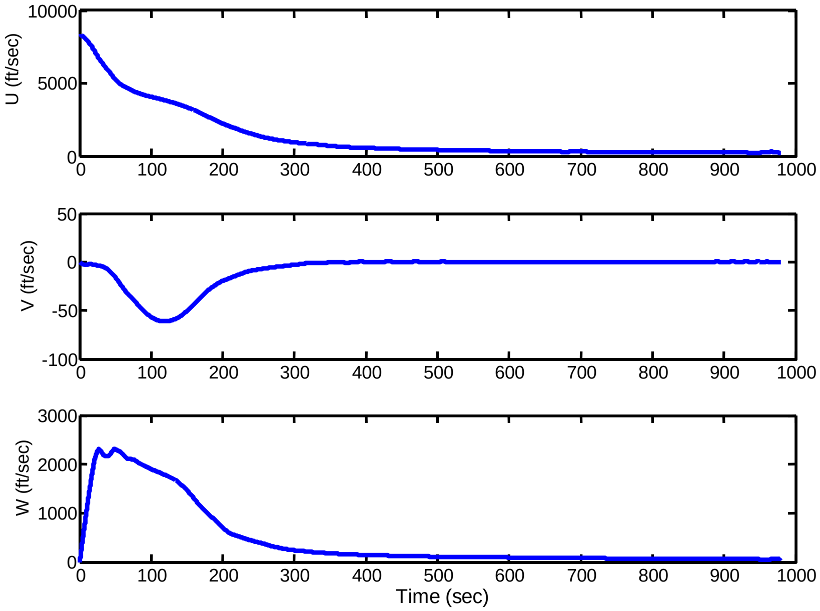


Figure 5.29 Max CR Linear Body-Relative Velocities (U,V,W).

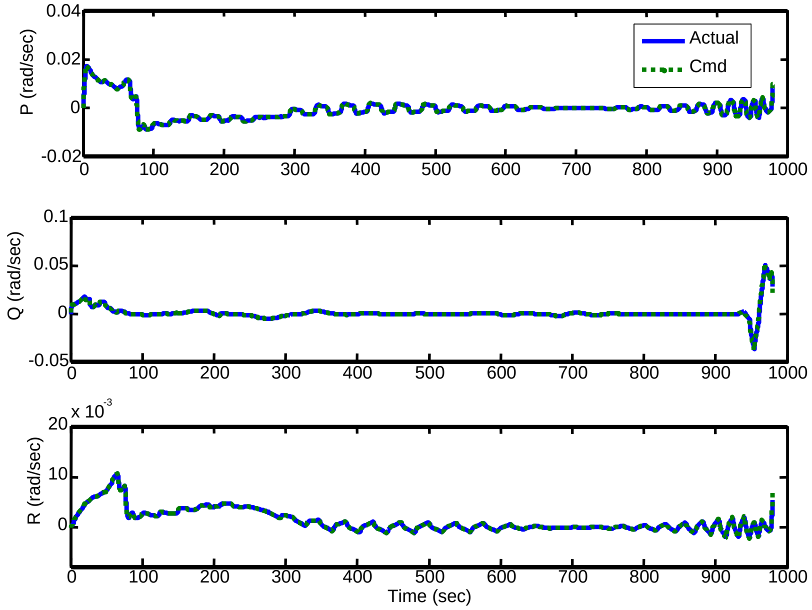


Figure 5.30   Max CR Angular Body-Rates (P,Q,R).

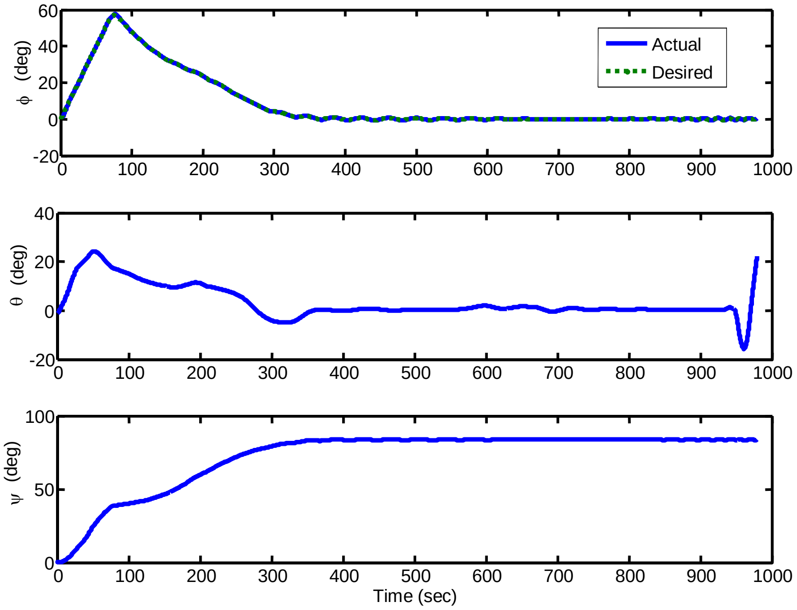


Figure 5.31   Max CR Euler Angles (Phi, Theta, Psi).

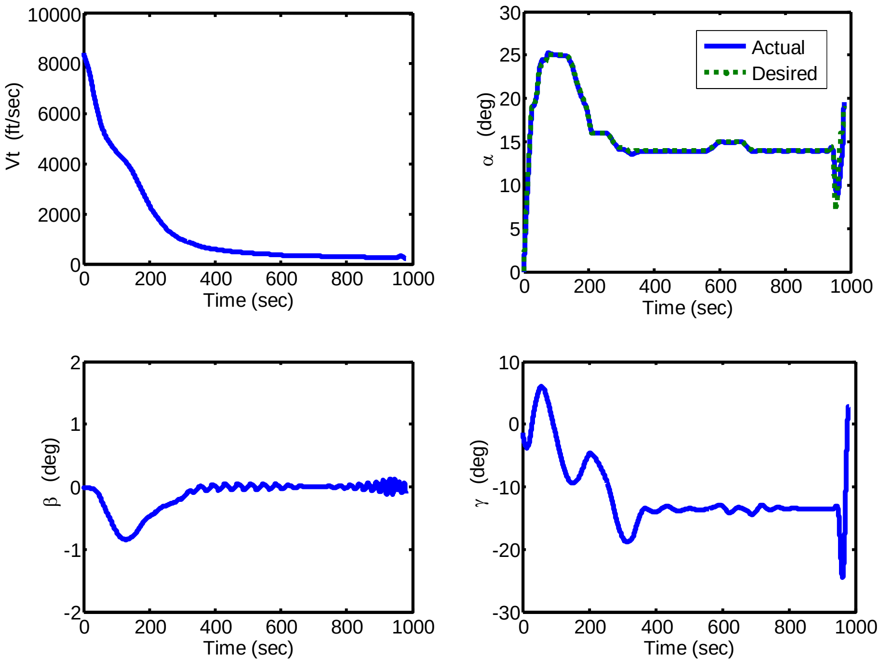


Figure 5.32 Max CR Airspeed (V) and Wind-Relative Angles (Alpha, Beta, Gamma).

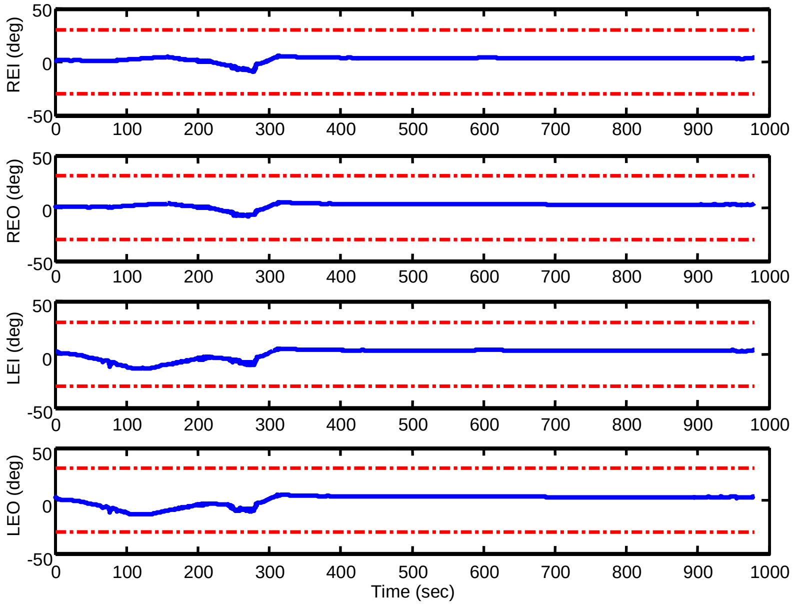


Figure 5.33 Max CR Elevon Control Surface Deflections.

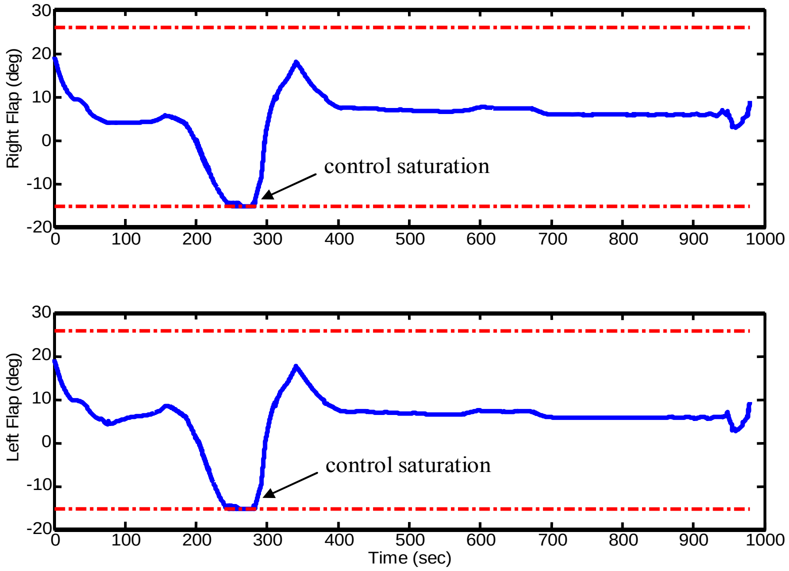


Figure 5.34 Max CR Body Flap Control Surface Deflections.

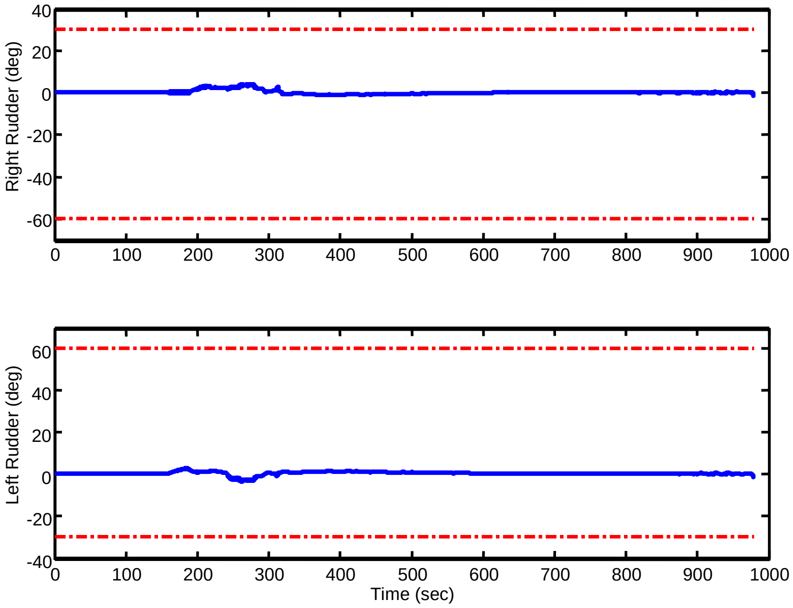


Figure 5.35 Max CR Rudder Control Surface Deflections.

Similar to the max DR case, Figure 5.33 to Figure 5.35 show that all of the control surface deflections in the 6-DOF simulation remain within their respective limits as indicated by the dash-dot lines. Again, as indicated in Figure 5.34, there is only one region of body-flap control saturation from approximately 240-280 sec. Also, Figure 5.36 shows that the anti-windup values remain approximately zero (10^{-15}) throughout the flight simulation.

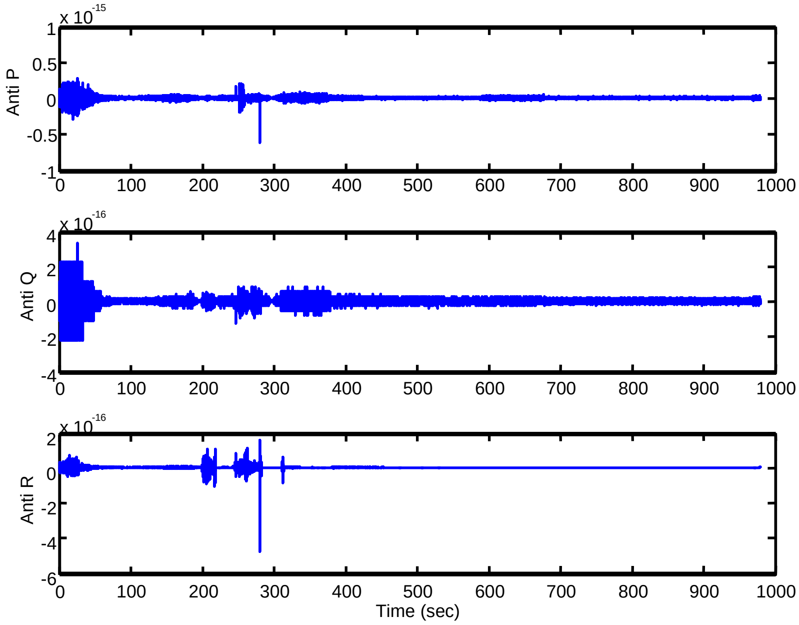


Figure 5.36 Max CR Anti-Windup Signals.

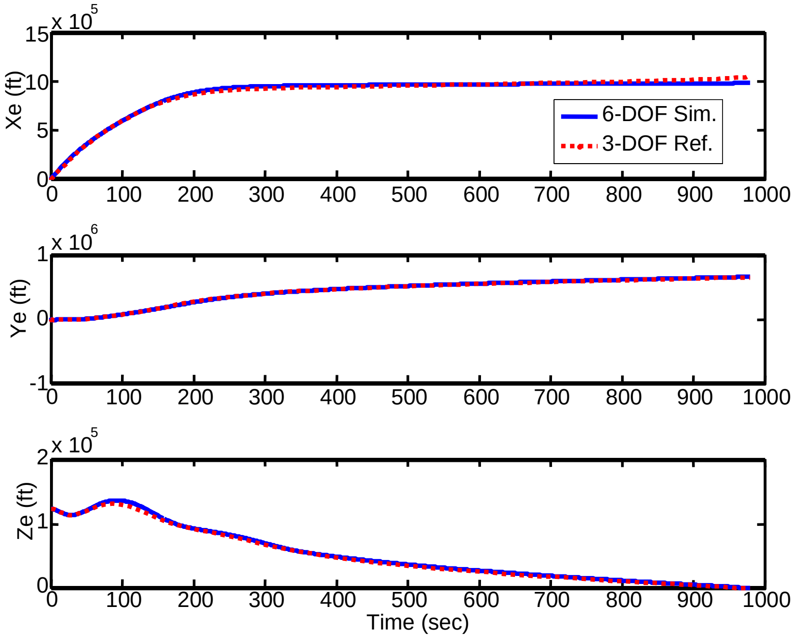


Figure 5.37 Max CR Comparison of Reference and Tracking Trajectory.

Figure 5.37 verifies that the guidance commands result in an acceptable reentry trajectory when comparing the 6-DOF simulation and the 3-DOF reference x,y,z-state histories. The corresponding forces and moments are shown in Figure 5.38 to Figure 5.43.

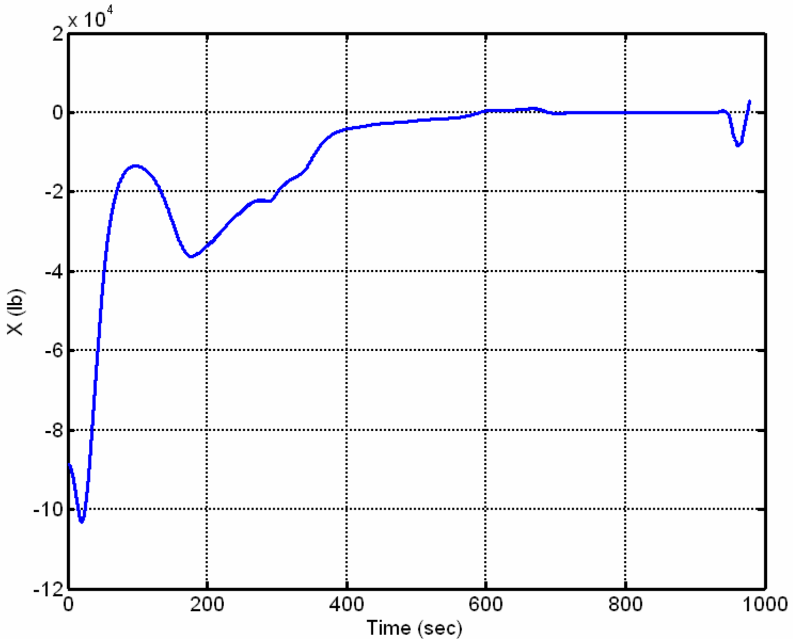


Figure 5.38   Axial Force History for Max CR Trajectory/

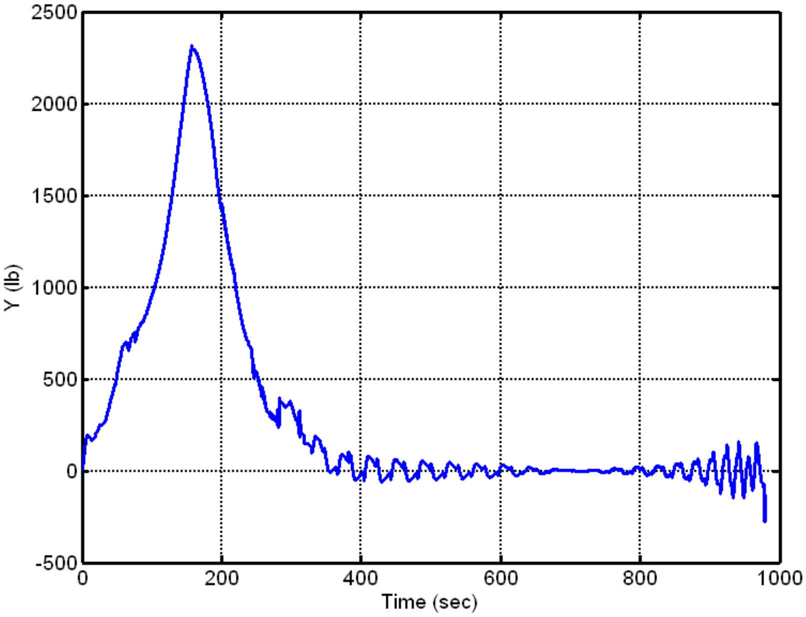


Figure 5.39   Side Force History for Max CR Trajectory.

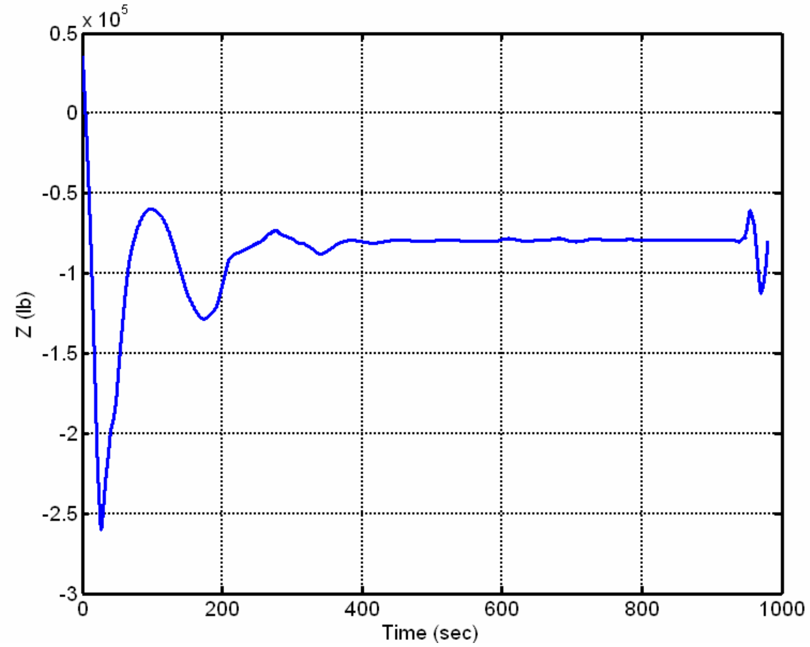


Figure 5.40   Normal Force History for Max CR Trajectory.

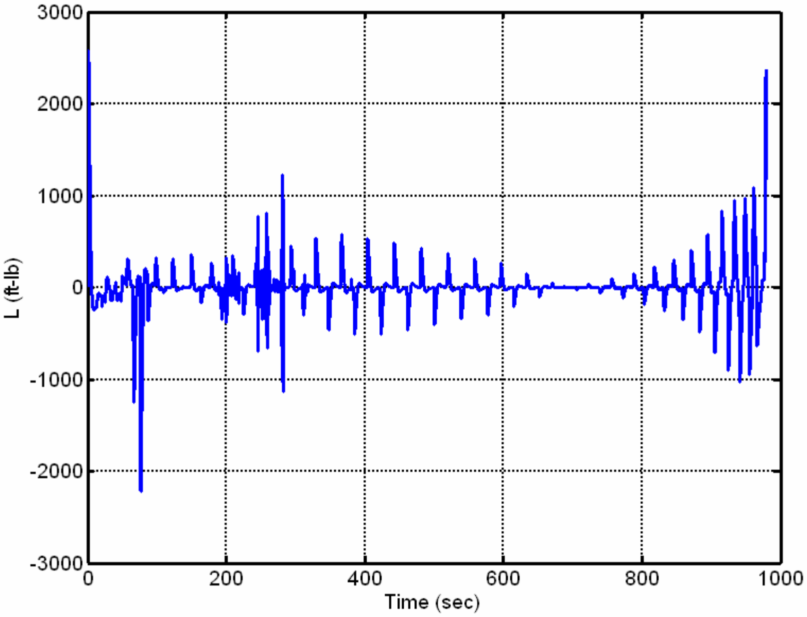


Figure 5.41   Rolling Moment History for Max CR Trajectory.

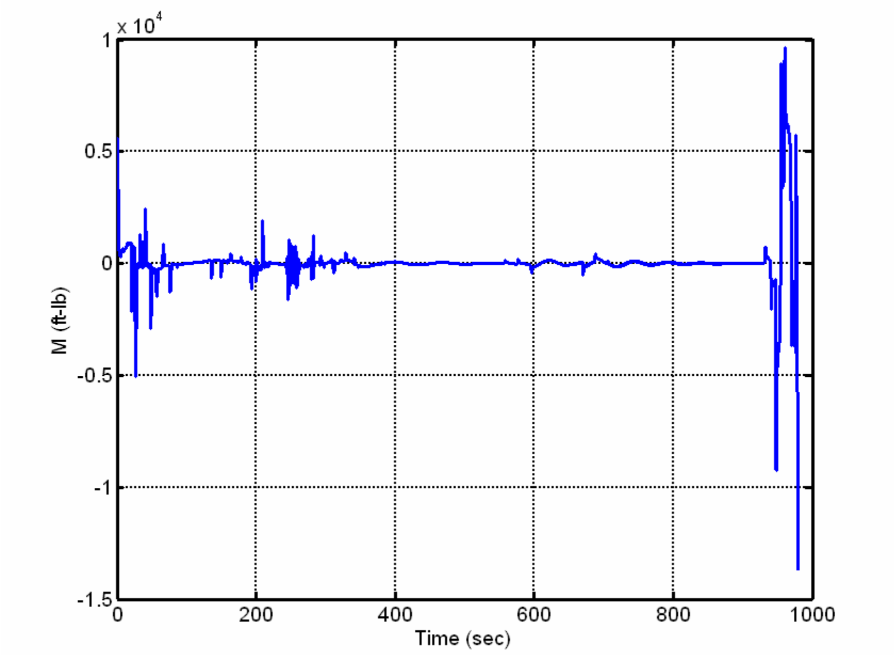


Figure 5.42   Pitching Moment History for Max CR Trajectory.

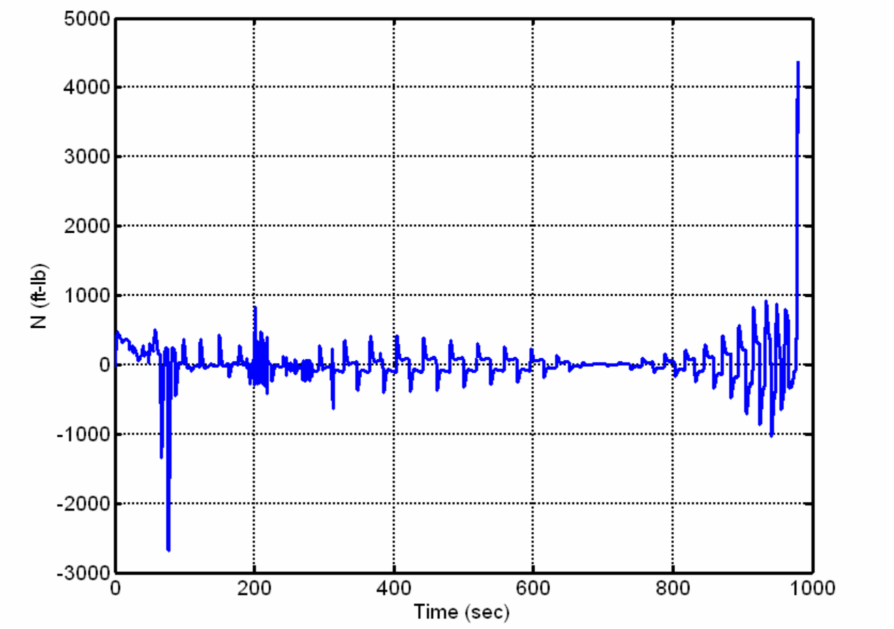


Figure 5.43   Yawing Moment History for Max CR Trajectory.

C. CONCLUDING REMARKS

This section presented the successful integration of a reconfigurable inner-loop control law consisting of DI, control allocation, model reference prefilters, and anti-windup integrators with an outer-loop, optimal guidance command generator. As demonstrated, the inner-loop control law was capable of tracking the body-frame angular rates that were converted from the wind-relative α and ϕ modulation of the off-line reference trajectory. Although the optimal trajectory generation was done off-line for this work, a similar model has already been demonstrated in Chapter IV to work for on-line reentry applications using the same Legendre PS method.

The potential drawback of the presented G&C architecture is that the inner-loop control system depends on various gains that require off-line tuning. This may limit the system's use for on-board autonomous applications, especially in situations involving unplanned maneuvers and/or flight anomalies. Even with the added robustness provided by the prefilters, anti-windup mechanism, and the reconfigurable control, additional modifications may still be needed to handle unexpected operational conditions. With this said, the next chapter omits the inner-loop controller and solves for the optimal control surface deflections directly. In this sense, the inner and outer loops are integrated into a single loop. As demonstrated by the 3-DOF results presented in Chapter IV, this eliminates the use of gains and all together providing a more robust and autonomous G&C system.

VI. 6-DOF TRAJECTORY OPTIMIZATION

A. INTRODUCTION

6-DOF modeling and simulation is typically used only in high-risk programs as a means to validate methods based on lower-order models. Traditionally, inner and outer loops are decoupled such that they are actually being solved separately. In this sense, the loops are decoupled. This premise is predominantly based on the assumption that control inputs to the outer loops are control variables for the inner loops and that the vehicle has instantaneous responses from the inner loops. In the aerospace industry, it is not uncommon that separate design teams independently develop guidance and control (G&C) methods. For example, there may be a separate team for developing high-level tasks embedded in guidance algorithms such as path planning whereas lower-level; control algorithms are developed by another team. In terms of classical control systems, the guidance is associated with an outer loop responsible for providing commands whereas the control is associated with an inner loop responsible for tracking the commands and actuating physical controls. In terms of motion, the guidance system provides translational motion of the vehicle's center-of-gravity (cg) whereas the control system provides rotational motion about the vehicle's cg. It is the goal of this work to explore the idea on departing from this conventional approach of separated G&C and adopting a more unified approach of integrated, *optimal* G&C.

1. Separated G&C Architecture

Consider a separated G&C architecture represented by the simplified block diagram shown in Figure 6.1. Here, the sole purpose of the guidance block is to generate a reference trajectory and the corresponding guidance commands based on high-level goals. Note that the guidance block as shown in Figure 6.1 is actually a combination of reference trajectory generation and guidance command generation as presented in the previous chapter. Typically, guidance command generation is independent of the control because it is assumed that the control system is well designed to track commands without delays. In many systems, the reference trajectories are generated off-line and then pre-programmed into the onboard computer's Flight Control System (FCS). This would be characterized as not having a "connected" outer loop for the trajectory generation.

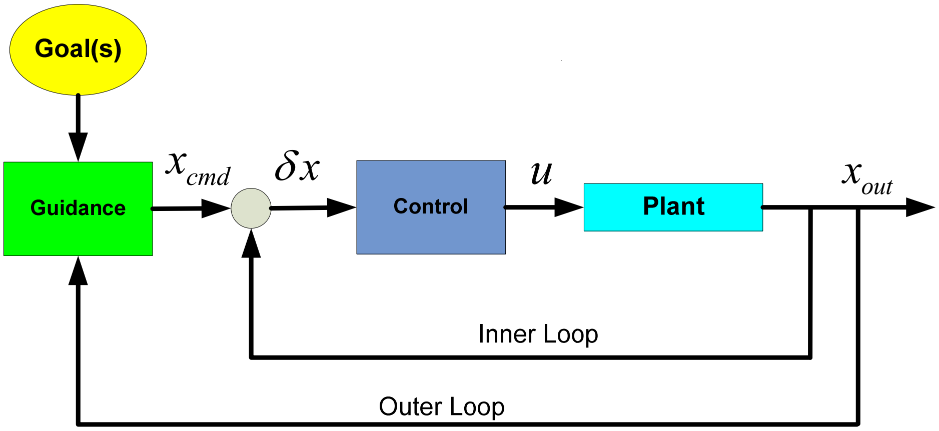


Figure 6.1 Separated G&C Architecture.

Another issue is that the two loops operate at different frequencies as a direct result of the different time constants associated with the translational and rotational dynamics. Therefore, depending on the dynamics of a specific system, the inner loop generally operates at much higher frequencies than the outer loop. For example, the frame rates of the inner-loop FCS for most RLV prototypes operate around 50 Hz whereas the outer loop operates around 10 Hz.

Under certain circumstances, it is possible that the inner-loop controller cannot track the guidance commands. This situation could be a result of an infeasible trajectory, limited control power, rate limited actuators, control failures, vehicle damage, etc. In the case of degraded inner-loop response, modification of the guidance loop gains is required (i.e., guidance adaptation). Under more extreme conditions that require retargeting and where guidance adaptation is insufficient, onboard trajectory reshaping is required.

In addition, employing a separated G&C architecture may potentially introduce problems as a result of conflicting objectives and/or design assumptions in the two loops. For example, Shaffer exposed a disconnect between the inner-loop control allocator and outer-loop guidance when solving an X-33 RLV reentry problem for footprint determination under nominal and control effector failure scenarios [46]. Incompatible cost functions between the inner-loop optimal control allocation problem and the outer-loop optimal guidance problem caused the failed vehicle to provide better performance than the nominal vehicle. This specific case was a result of naively defining the control allocator cost function to minimize control surface deflections for a minimum-range case. Perhaps a more appropriate inner-loop cost would have been to configure the surfaces for maximum control surface deflections or minimum change in L/D to be consistent with the overall objective of minimizing downrange. Regardless, this example pointed out the potential problems that can occur from the natural disconnect between the two loops. In addition to the objective conflicts, there could be additional underlying sources of error. For example, the inner-loop control algorithm uses a piecewise linear programming (PLP) method to approximate the nonlinear optimal control allocation whereas the outer-loop guidance uses a direct optimal control method to solve a full, nonlinear system posed in a nonlinear programming (NLP) fashion.

2. Integrated G&C Architecture

One way to alleviate most of these problems is by employing an architecture that integrates the functions of the G&C into a single loop as illustrated in Figure 6.2. From a practical standpoint, this may consist of pushing the outer-loop down to a lower-level in the control architecture.

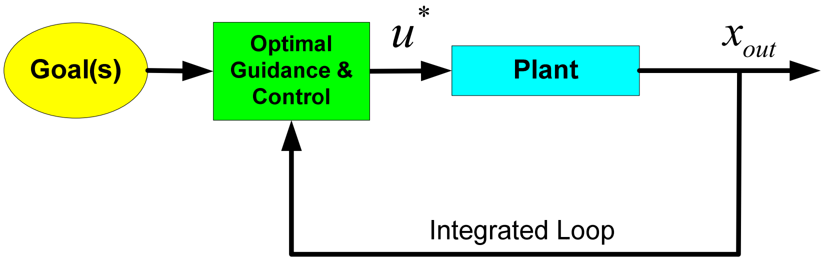


Figure 6.2 Integrated G&C Architecture.

Most research work uses 3-DOF models since they capture the majority of the translational effects, using α, β-modulation as controls; however, to emphasize attitude dynamics and capture translational and rotational effects, higher-order models are required. As detailed in Chapter III, it is well known that model fidelity matters; however, it is often overlooked or taken for granted. Demonstrated by the work of Fahroo et al., model fidelity has an obvious impact on performance footprints [85]-[87]. For full 6-DOF models, the aerodynamic forces and moments are untrimmed and depend directly on the control surfaces. Hence, by directly solving for the control surface deflections using an "integrated" G&C approach, the highly-coupled nature of translational and rotational motion should be more accurately accounted for, consequently providing a more accurate representation of the true vehicle performance.

Also, to help achieve the goal of intelligent, autonomous G&C, the dependence on time-consuming gain tuning/scheduling, trial-and-error, linearization, trim techniques, etc, must be removed. In other words, less human interaction and less off-line analysis is required. Solving the full, nonlinear system in its purest form using optimal control techniques is the approach pursued in this chapter. In general, "integrated" 6-DOF G&C *should* accomplish the following:

1. Reveal effects associated with decoupled inner/outer loops
2. Remove the need for inner-loop tracking in the traditional sense; hence, no tracking errors nor undesired behavior such as tracking delays
3. Improve "footprint" generation via more accurate solutions based on a model that captures the full physics of the problem without unnecessary assumptions or constraints

Transitioning from the intermediary work presented in Chapter V, it is the intent of this chapter to explore some of these concerns by solving the full 6-DOF optimal control problem in the framework of the single, integrated G&C architecture.

B. 6-DOF OPTIMAL TRAJECTORY GENERATION

Before completely abandoning the optimal tracking approach as presented in Chapter V, it is first necessary to evaluate the suitability and effectiveness of the PS method in solving the full, 6-DOF optimal control problem for the reentry of the X-33 vehicle. This problem entails the simultaneous optimization of eight controls embedded in the high-fidelity (HiFi) translational and rotational equations of motion (EoM).

1. 6-DOF Model

Studying the effectiveness of this new "integrated" G&C approach includes the experimentation of using different models and problem formulations. Instead of introducing each set of equations in this section, as done for the 3-DOF chapter, it is more logical to present each model with its corresponding problem formulation and results.

As in the 3-DOF problem formulation of Chapter IV, the 6-DOF problem formulation assumes the RLV is a rigid, lifting-body, and gliding unpowered back to earth from a suborbital altitude. For each of the proceeding problem formulations, the objective function is either to maximize the range performance of the vehicle or minimize the miss distance from the center of a pre-determined final approach corridor to a designated landing site. Consistent with the nonlinear OCP formulation (see Chapter II), the objective function is subject to both equality and inequality constraints (i.e., state, control, event, path) that are defined for each problem. Although numerous problem formulations were studied in the course of this research, including various degrees of model fidelity, only the most relevant ones are presented in this section.

2. Problem SF: Standard Flat-Earth Model

a. Problem Formulation

Since the original intent of this work was to investigate control surface failure scenarios using the 6-DOF, optimal "integrated" G&C method, the initial condition was selected such that spherical, rotating earth effects and variations in gravity would have negligible effect. As such, this limited scope facilitated the use of the standard "flat-earth" equations as defined in Chapter III (see Fig. 3.13). Recall that these equations assume a Cartesian "local horizon" coordinate system represented by the typical 3-D positional elements, (x, y, z) whereby both the translational and rotational equations are with respect to a body-fixed reference frame. As detailed in Chapter III, the 6-DOF EoM are nonlinear and highly coupled through the various flight angle relations and the aerodynamic forces and moments. For this model, the aerodynamic force and moment coefficients are determined from the use of an extensive X-33 aerodynamic database. Recall that each coefficient has a functional dependency on angle-of-attack (α), sideslip angle (β), Mach number (M) which is essentially a function of velocity and speed-of-sound at a given altitude, $M = V / a(h)$, where $h \equiv z$, and the various control surface deflections represented by the effector vector,

$\underline{\delta} = [\delta_{RE_{in}}, \delta_{RE_{out}}, \delta_{LE_{in}}, \delta_{LE_{out}}, \delta_{RF}, \delta_{LF}, \delta_{RR}, \delta_{LR}]$. In the component build-up form, the aerodynamic coefficients for the X-33 are represented as shown in Figure 6.4 and Figure 6.4.

$$\begin{aligned}
\mathrm{C_A} = {} & \mathrm{C}_{\mathrm{A}_{\text{base}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{A}_{\hat{\mathrm{q}}}}(\alpha,M(V,h),q) + \mathrm{C}_{\mathrm{A}_{\delta_{REi}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{A}_{\delta_{REo}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{A}_{\delta_{LEi}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{A}_{\delta_{LEo}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{A}_{\delta_{RF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{A}_{\delta_{LF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{A}_{\delta_{RR}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{A}_{\delta_{LR}}}(\alpha,\beta,M(V,h))
\end{aligned}$$

$$\begin{aligned}
\mathrm{C_Y} = {} & \mathrm{C}_{\mathrm{Y}_{\text{base}}}(\beta) + \mathrm{C}_{\mathrm{Y}_{\hat{\mathrm{p}}}}(\alpha,M(V,h),p) + \mathrm{C}_{\mathrm{Y}_{\hat{\mathrm{r}}}}(\alpha,M(V,h),r) + \mathrm{C}_{\mathrm{Y}_{\delta_{REi}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{Y}_{\delta_{REo}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{Y}_{\delta_{LEi}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{Y}_{\delta_{LEo}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{Y}_{\delta_{RF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{Y}_{\delta_{LF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{Y}_{\delta_{RR}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{Y}_{\delta_{LR}}}(\alpha,\beta,M(V,h))
\end{aligned}$$

$$\begin{aligned}
\mathrm{C_N} = {} & \mathrm{C}_{\mathrm{N}_{base}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{N}_{\hat{q}}}(\alpha,M(V,h),q) + \mathrm{C}_{\mathrm{N}_{\delta_{REi}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{N}_{\delta_{REo}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{N}_{\delta_{LEi}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{N}_{\delta_{LEo}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{N}_{\delta_{RF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{N}_{\delta_{LF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{N}_{\delta_{RR}}}(\alpha,\beta,M(V,h)) \\
& + \mathrm{C}_{\mathrm{N}_{\delta_{LR}}}(\alpha,\beta,M(V,h))
\end{aligned}$$

Figure 6.3 Aerodynamic Force Coefficients in the Component Build-Up Form.

$$\boxed{\begin{aligned}
\mathrm{C_l} &= \mathrm{C}_{\mathrm{l_{base}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{l}_{\hat{\mathrm{r}}}}(\alpha,M(V,h),r) + \mathrm{C}_{\mathrm{l}_{\hat{\mathrm{p}}}}(\alpha,M(V,h),p) \\
&\quad + \mathrm{C}_{\mathrm{l}_{\delta_{REi}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{l}_{\delta_{REo}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{l}_{\delta_{LEi}}}(\alpha,\beta,M(V,h)) \\
&\quad + \mathrm{C}_{\mathrm{l}_{\delta_{LEo}}}(\alpha,\beta,M(V,h)) + C_{\mathrm{l}_{\delta_{RF}}}(\alpha,\beta,M(V,h)) + C_{\mathrm{l}_{\delta_{LF}}}(\alpha,\beta,M(V,h)) \\
&\quad + C_{\mathrm{l}_{\delta_{RR}}}(\alpha,\beta,M(V,h)) + C_{\mathrm{l}_{\delta_{LR}}}(\alpha,\beta,M(V,h)) \\
\\
\mathrm{C_m} &= \mathrm{C}_{\mathrm{m_{base}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{m}_{\hat{\mathrm{q}}}}(\alpha,M(V,h),q) + \mathrm{C}_{\mathrm{m}_{\delta_{REi}}}(\alpha,\beta,M(V,h)) \\
&\quad + \mathrm{C}_{\mathrm{m}_{\delta_{REo}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{m}_{\delta_{LEi}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{m}_{\delta_{LEo}}}(\alpha,\beta,M(V,h)) \\
&\quad + \mathrm{C}_{\mathrm{m}_{\delta_{RF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{m}_{\delta_{LF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{m}_{\delta_{RR}}}(\alpha,\beta,M(V,h)) \\
&\quad + \mathrm{C}_{\mathrm{m}_{\delta_{LR}}}(\alpha,\beta,M(V,h)) \\
\\
\mathrm{C_n} &= \mathrm{C}_{\mathrm{n_{base}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{n}_{\hat{\mathrm{r}}}}(\alpha,M(V,h),r) + \mathrm{C}_{\mathrm{n}_{\hat{\mathrm{p}}}}(\alpha,M(V,h),p) \\
&\quad + \mathrm{C}_{\mathrm{n}_{\delta_{REi}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{n}_{\delta_{REo}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{n}_{\delta_{LEi}}}(\alpha,\beta,M(V,h)) \\
&\quad + \mathrm{C}_{\mathrm{n}_{\delta_{LEo}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{n}_{\delta_{RF}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{n}_{\delta_{LF}}}(\alpha,\beta,M(V,h)) \\
&\quad + \mathrm{C}_{\mathrm{n}_{\delta_{RR}}}(\alpha,\beta,M(V,h)) + \mathrm{C}_{\mathrm{n}_{\delta_{RR}}}(\alpha,\beta,M(V,h))
\end{aligned}}$$

Figure 6.4 Aerodynamic Moment Coefficients in the Component Build-Up Form.

For this problem, the state and control vectors are defined as

$$\begin{aligned}
\underline{x} &= [x_E, y_E, z_E, V, \alpha, \beta, p, q, r, \phi, \theta, \psi]^T \in \mathbb{R}^{12} \\
\underline{u} = \underline{\delta} &= [\delta_{RE_{in}}, \delta_{RE_{out}}, \delta_{LE_{in}}, \delta_{LE_{out}}, \delta_{RF}, \delta_{LF}, \delta_{RR}, \delta_{LR}]^T \in \mathbb{R}^{8} \\
&\equiv [u_1, \ldots, u_8]
\end{aligned} \tag{5.36}$$

Here the control vector assumes there are no command delays as discussed earlier. To model the rates, an alternate formulation uses the "virtual" control vector

$$\underline{u} = u_{\dot{\underline{\delta}}} = [u_{\dot{\delta}_{RE_{in}}}, u_{\dot{\delta}_{RE_{out}}}, u_{\dot{\delta}_{LE_{in}}}, u_{\dot{\delta}_{LE_{out}}}, u_{\dot{\delta}_{RF}}, u_{\dot{\delta}_{LF}}, u_{\dot{\delta}_{RR}}, u_{\dot{\delta}_{LR}}]^T \in \mathbb{R}^{8} \tag{5.37}$$

This requires the modification of the original state vector to include the physical controls, $\underline{\delta}$, such that

$$\underline{x} = [x_E, y_E, z_E, V, \alpha, \beta, p, q, r, \phi, \theta, \psi, \delta_{RE_{in}}, \delta_{RE_{out}}, \delta_{LE_{in}}, \delta_{LE_{out}}, \delta_{RF}, \delta_{LF}, \delta_{RR}, \delta_{LR}]^T \in \mathbb{R}^{20} \tag{5.38}$$

For now, this method of using "virtual" controls (i.e., pseudo-controls), is only used if the original formulation results in a physically unrealizable, non-smooth control solution.

Now, to maximize the range, the cost function is subject to the following constraints.

Dynamic Constraints, $\dot{x} = f(x(t),u(t),t)$:

$$\underline{\dot{x}} = [\dot{x}_E, \dot{y}_E, \dot{z}_E, \dot{V}, \dot{\alpha}, \dot{\beta}, \dot{p}, \dot{q}, \dot{r}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T = f(x,u,t) \tag{5.39}$$

where $f(x,u,t)$ is the right-hand-side (RHS) of the dynamic EoM specified in Figure 3.13 (see Chapter III).

Path Constraints, $h^L \le h(x(t),u(t),t) \le h^U$:

$$\begin{bmatrix} -2.5\,\text{g's} \\ 0 \\ 0 \end{bmatrix} \le \begin{bmatrix} n_z(z,V,\alpha) \\ \overline{q}(z,V) \\ Q(z,V) \end{bmatrix} \le \begin{bmatrix} 2.5\,\text{g's} \\ 300\,\text{lb/ft}^2 \\ 60\,\text{BTU/ft-s} \end{bmatrix} \tag{5.40}$$

Initial and Final Endpoint Conditions, $e^L \le e(x(t_0), x(t_f), t_0, t_f) \le e^U$:

$$\begin{array}{llll} \text{x}(\text{t}_0) = 0\text{ ft} & \text{V}(\text{t}_0) = 5413\text{ ft/s} & \text{p}(t_0) = 0\,\text{deg/s} & \phi(t_0) = 0\,\text{deg} \\ \text{y}(\text{t}_0) = 0\text{ ft} & \alpha(t_0) = 19\text{ deg} & \text{q}(t_0) = 0\,\text{deg/s} & \theta(t_0) = \text{-2 deg} \\ \text{z}(\text{t}_0) = 125\text{e}3\text{ ft} & \beta(t_0) = 0\text{ deg} & \text{r}(t_0) = 0\,\text{deg/s} & \psi(t_0) = 0\text{ deg} \end{array} \tag{5.41}$$

$$\begin{array}{l} z(t_f) = 500\,\text{ft} \\ V(t_f) = 335.13\,\text{ft/s} \end{array} \tag{5.42}$$

$$-25\,\text{ft/s} \le \dot{z}(t_f) \le 8.33\,\text{ft/s} \tag{5.43}$$

with the following initial and final guesses (linear interpolation between two points)

$$
\begin{array}{ll}
\mathrm{x}_0 = 0 \text{ ft} & \mathrm{x}_\mathrm{f} = 1e6 \text{ ft} \\
\mathrm{y}_0 = 0 \text{ ft} & \mathrm{y}_\mathrm{f} = 0 \text{ ft} \\
\mathrm{z}_0 = 125\mathrm{e}3 \text{ ft} & \mathrm{z}_\mathrm{f} = 500 \text{ ft} \\
\mathrm{V}_0 = 5413 \text{ ft/s} & \mathrm{V}_\mathrm{f} = 335.13 \text{ ft/s} \\
\alpha_0 = 19 \text{ deg} & \alpha_\mathrm{f} = 0 \text{ deg} \\
\beta_0 = 0 \text{ deg} & \beta_\mathrm{f} = 0 \text{ deg} \\
\mathrm{p}_0 = 0 \text{ deg/s} & \mathrm{p}_\mathrm{f} = 0 \text{ deg/s} \\
\mathrm{q}_0 = 0 \text{ deg/s} & \mathrm{q}_\mathrm{f} = 0 \text{ deg/s} \\
\mathrm{r}_0 = 0 \text{ deg/s} & \mathrm{r}_\mathrm{f} = 0 \text{ deg/s} \\
\phi_0 = 0 \text{ deg} & \phi_\mathrm{f} = 0 \text{ deg} \\
\theta_0 = \text{-2 deg} & \theta_\mathrm{f} = 0 \text{ deg} \\
\psi_0 = 0 \text{ deg} & \psi_\mathrm{f} = 0 \text{ deg}
\end{array}
\tag{5.44}
$$

$$
\begin{array}{ll}
\delta_{\mathrm{REi}_0} = \text{-0.8 deg} & \delta_{\mathrm{REi}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{REo}_0} = \text{-0.8 deg} & \delta_{\mathrm{REo}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{LEi}_0} = \text{-0.8 deg} & \delta_{\mathrm{LEi}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{LEo}_0} = \text{-0.8 deg} & \delta_{\mathrm{LEo}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{RF}_0} = 7.9 \text{ deg} & \delta_{\mathrm{RF}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{LF}_0} = 7.9 \text{ deg} & \delta_{\mathrm{LF}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{RR}_0} = 0 \text{ deg} & \delta_{\mathrm{RR}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{LR}_0} = 0 \text{ deg} & \delta_{\mathrm{LR}_\mathrm{f}} = 0 \text{ deg}
\end{array}
\tag{5.45}
$$

Note that the initial control surface deflections selected were based on a trim solution for equilibrium glide at h=125,000 ft, V=5413 ft/s, and α =19 deg.

State Variable Constraints, $x^L \le x(t) \le x^U$:

$$
\begin{array}{ll}
0 \text{ ft} \le x_E \le 3e6 \text{ ft} & -30 \deg/\mathrm{s} \le p \le 30 \deg/\mathrm{s} \\
-1e6 \text{ ft} \le y_E \le 1e6 \text{ ft} & -30 \deg/\mathrm{s} \le q \le 30 \deg/\mathrm{s} \\
0 \text{ ft} \le z_E \le 260e3 \text{ ft} & -30 \deg/\mathrm{s} \le r \le 30 \deg/\mathrm{s} \\
1.0 \text{ ft/s} \le V \le 2 * V(t_0) \text{ ft/s} & -90 \deg \le \phi \le 90 \deg \\
-10 \deg \le \alpha \le 50 \deg & -89 \deg \le \theta \le 89 \deg \\
-1.5 \deg \le \beta \le 1.5 \deg & -180 \deg \le \psi \le 180 \deg
\end{array}
\tag{5.46}
$$

and Control Variable Constraints, $u^L \le u(t) \le u^U$:

$$\begin{aligned}
&-30\deg \le \delta_i \le 30\deg && i = 1,2,3,4 \\
&-15\deg \le \delta_j \le 26\deg && j = 5,6 \\
&-60\deg \le \delta_7 \le 30\deg && \\
&-30\deg \le \delta_8 \le 60\deg &&
\end{aligned} \tag{5.47}$$

According to Ref. [194], the X-33 design doesn't allow for the rudder controls to become active until $M \le 2.5$; therefore, the constraints are more appropriately defined as

$$\begin{aligned}
&-30\deg \le \delta_i \le 30\deg && i = 1,2,3,4 \\
&-20\deg \le \delta_j \le 20\deg && j = 5,6
\end{aligned} \tag{5.48}$$

$$\delta_{7,8} \begin{cases} = 0 & \text{if } \ M(V, z_E) > 2.5 \\ -30 \le \delta_{7,8} \le 30 & \text{if } \ M(V, z_E) \le 2.5 \end{cases} \tag{5.49}$$

where $M(V, z_E) = \dfrac{V}{a(z_E)}$. Note that some experimentation was performed using the rudder constraint given by Eq. (5.49); however, most of the results included in this dissertation omit this "hard" constraint to see how it effects the system, if at all, or if the physics of the problem naturally satisfies this constraint.

b. Results and Analysis

The following results were generated by bootstrapping from 14 to 28 nodes.

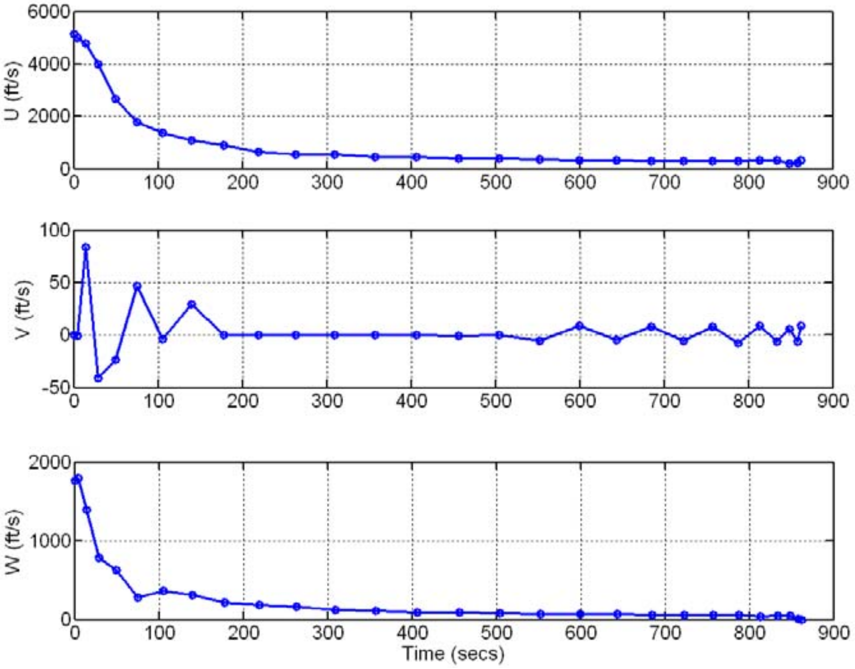


Figure 6.5 Max DR Linear Body-Relative Velocities.

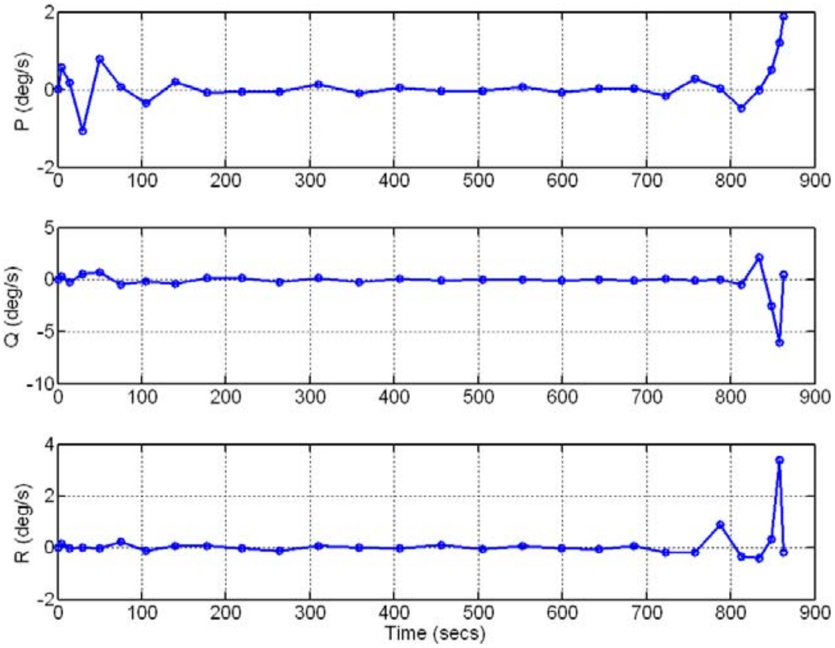


Figure 6.6 Max DR Angular Body-Rates.

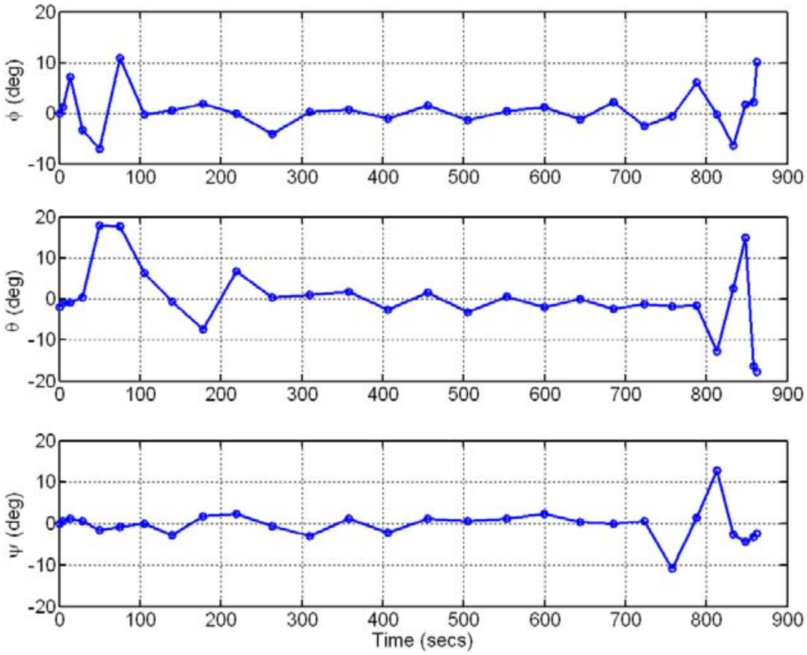


Figure 6.7 Max DR Euler Angles.

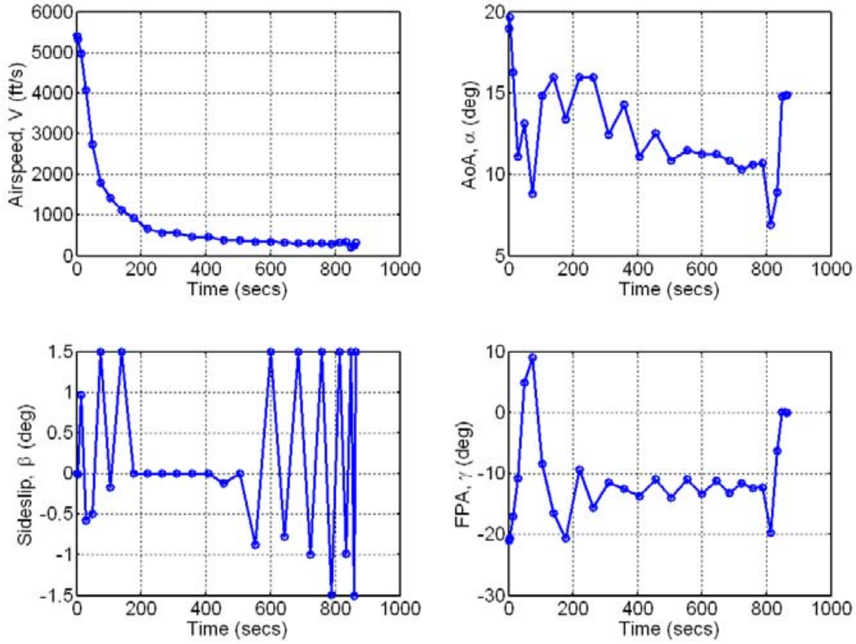


Figure 6.8 Max DR Airspeed and Wind-Relative Flight Angles.

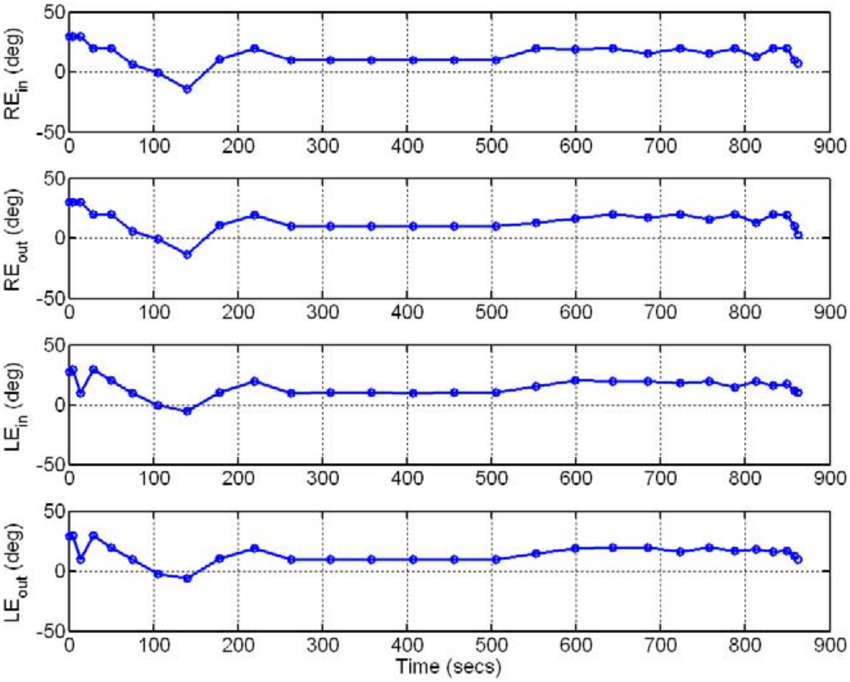


Figure 6.9 Max DR Elevon Controls.

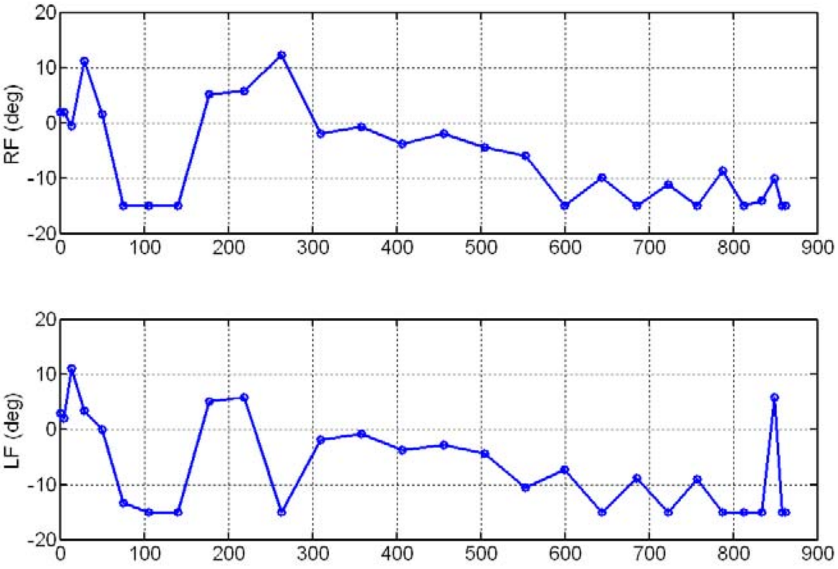


Figure 6.10 Max DR Flap Controls.

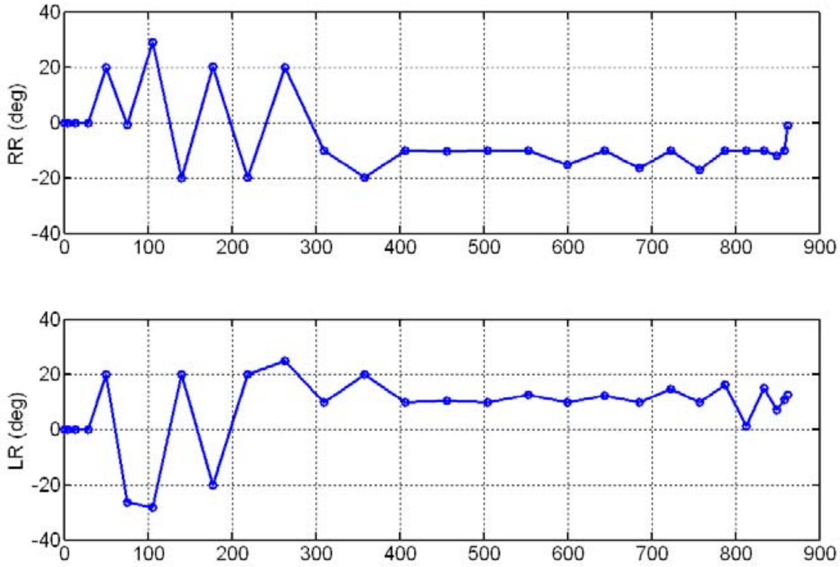


Figure 6.11 Max DR Rudder Controls.

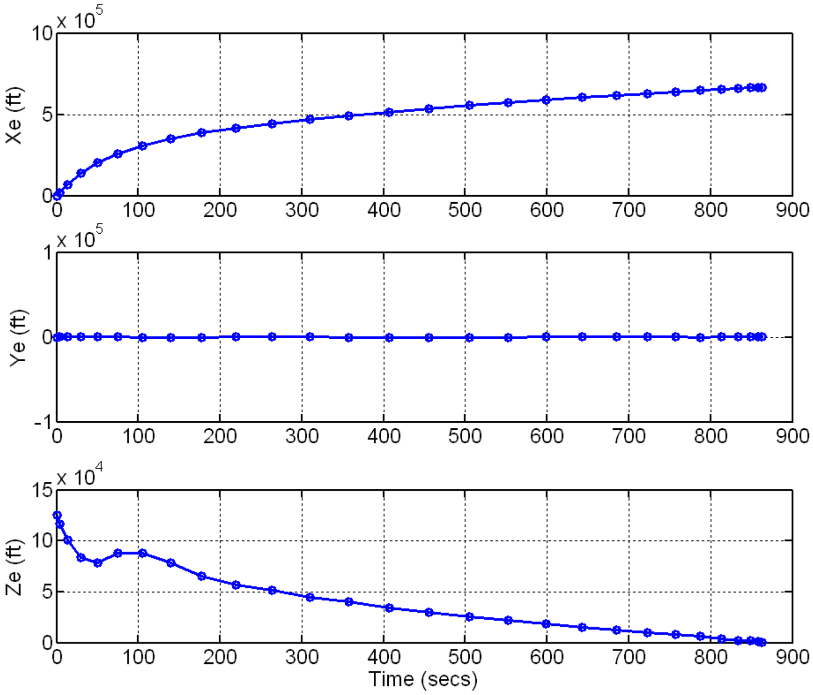


Figure 6.12 Max DR Cartesian Position.

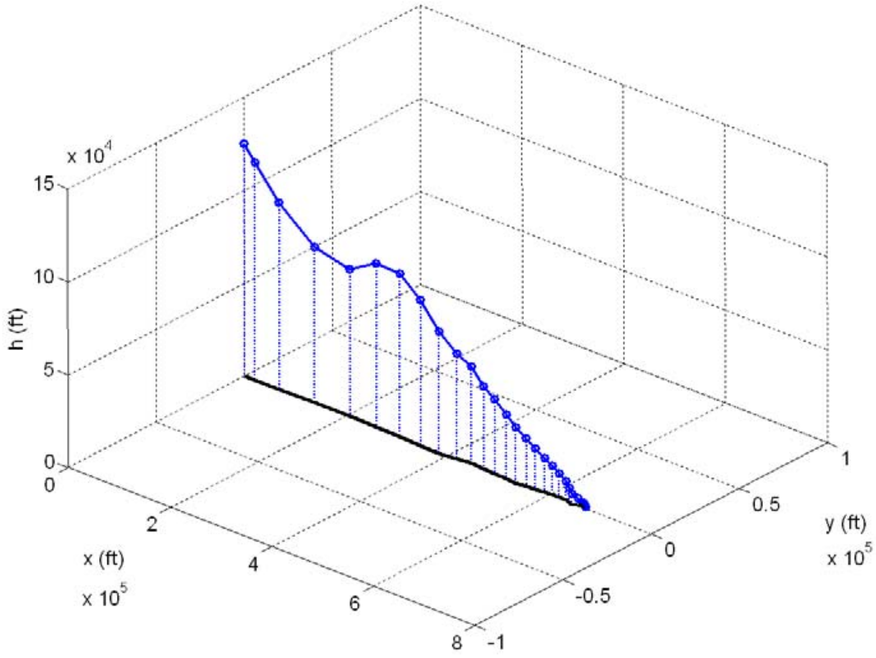


Figure 6.13 Max DR 3D Flight Trajectory Profile.

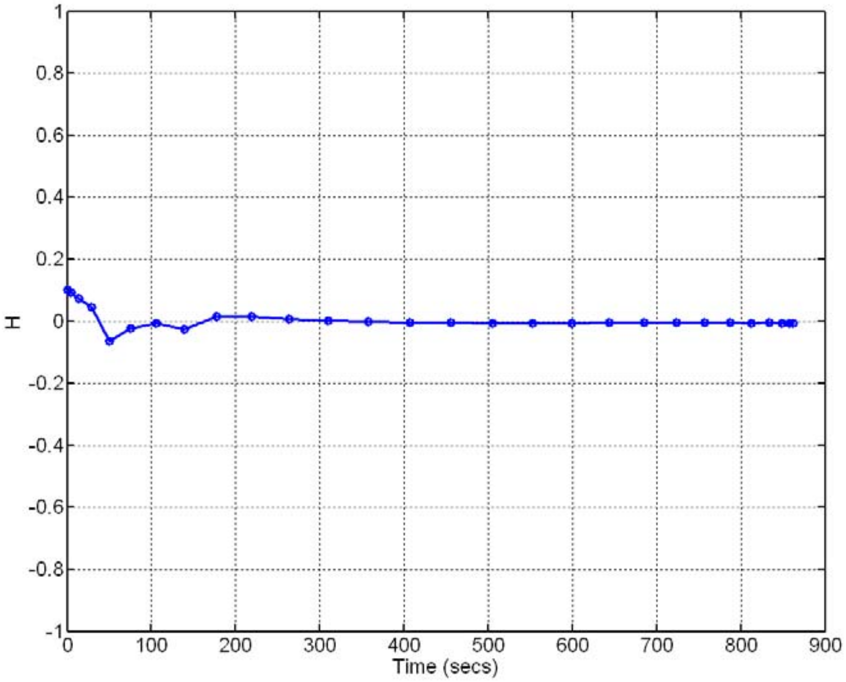


Figure 6.14 Max DR Hamiltonian Function.

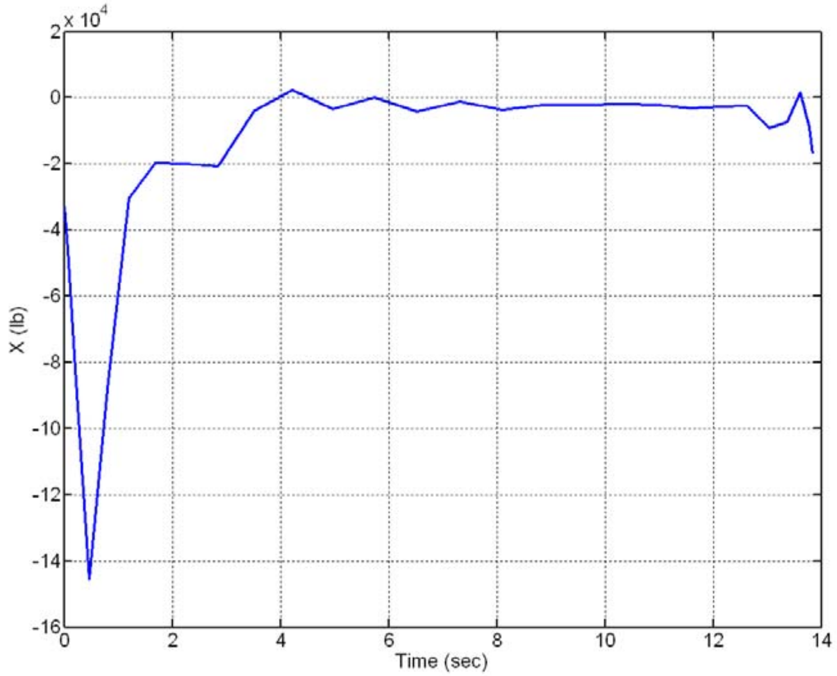


Figure 6.15   Max DR Axial Force History.

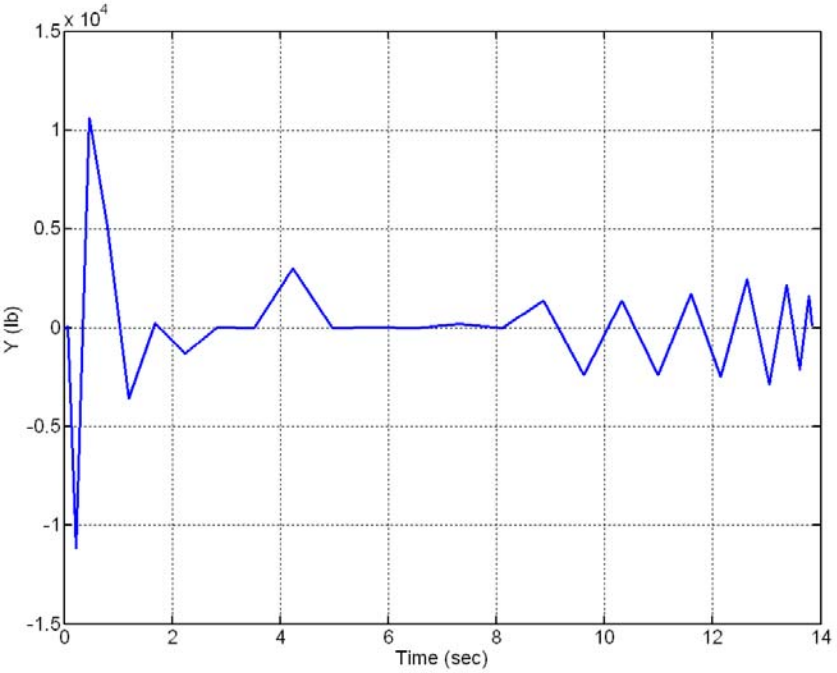


Figure 6.16   Max DR Side Force History.

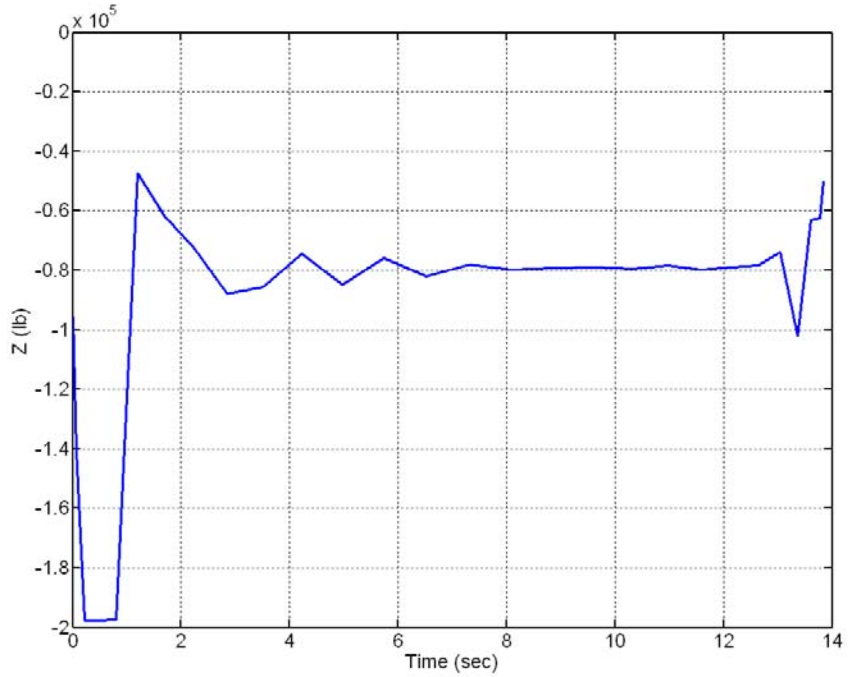


Figure 6.17 Max DR Normal Force History.

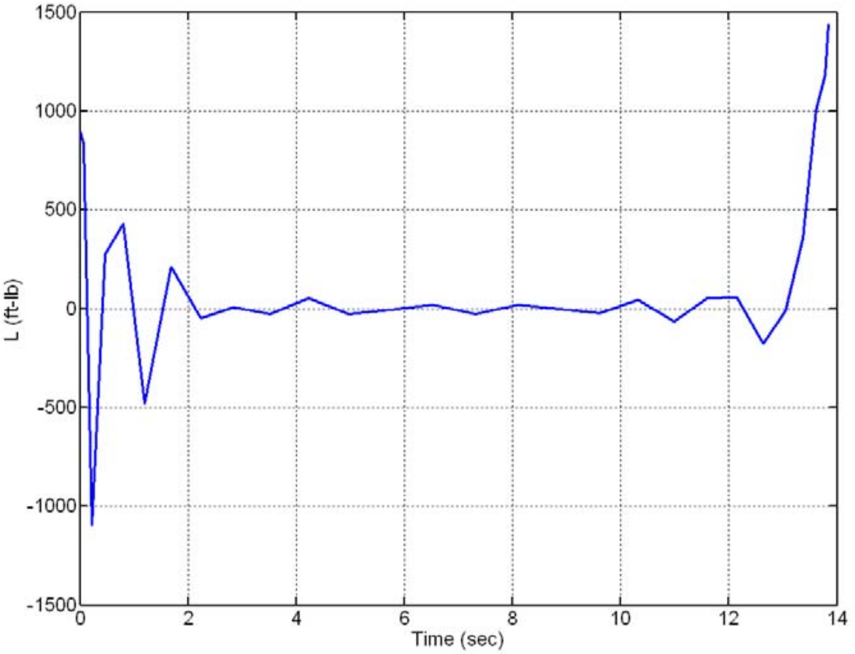


Figure 6.18 Max DR Rolling Moment History.

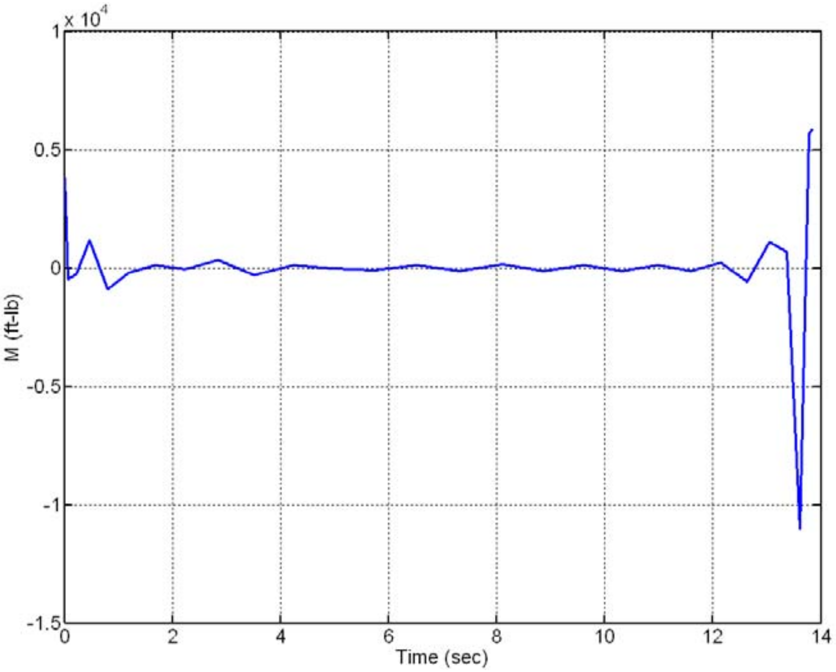


Figure 6.19 Max DR Pitching Moment History.

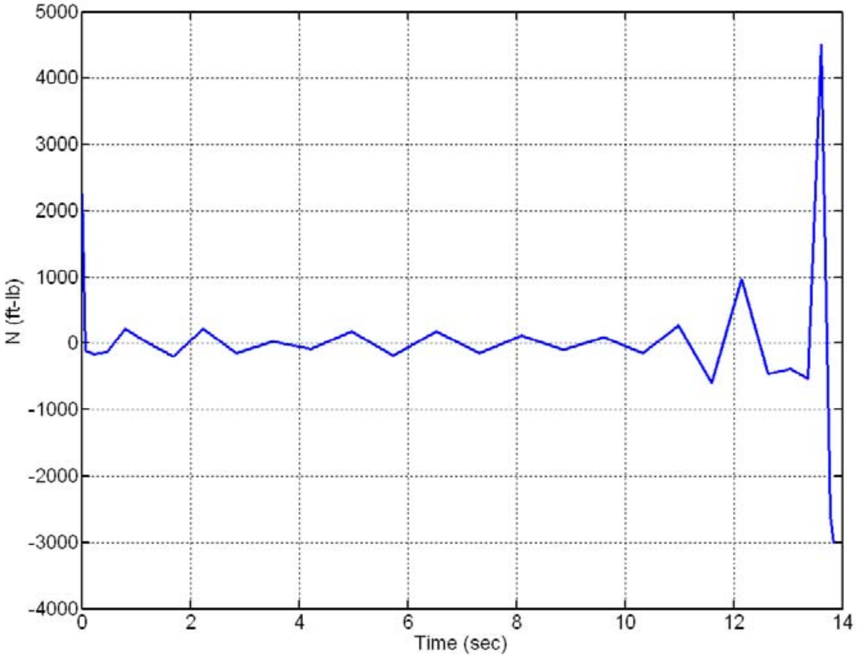
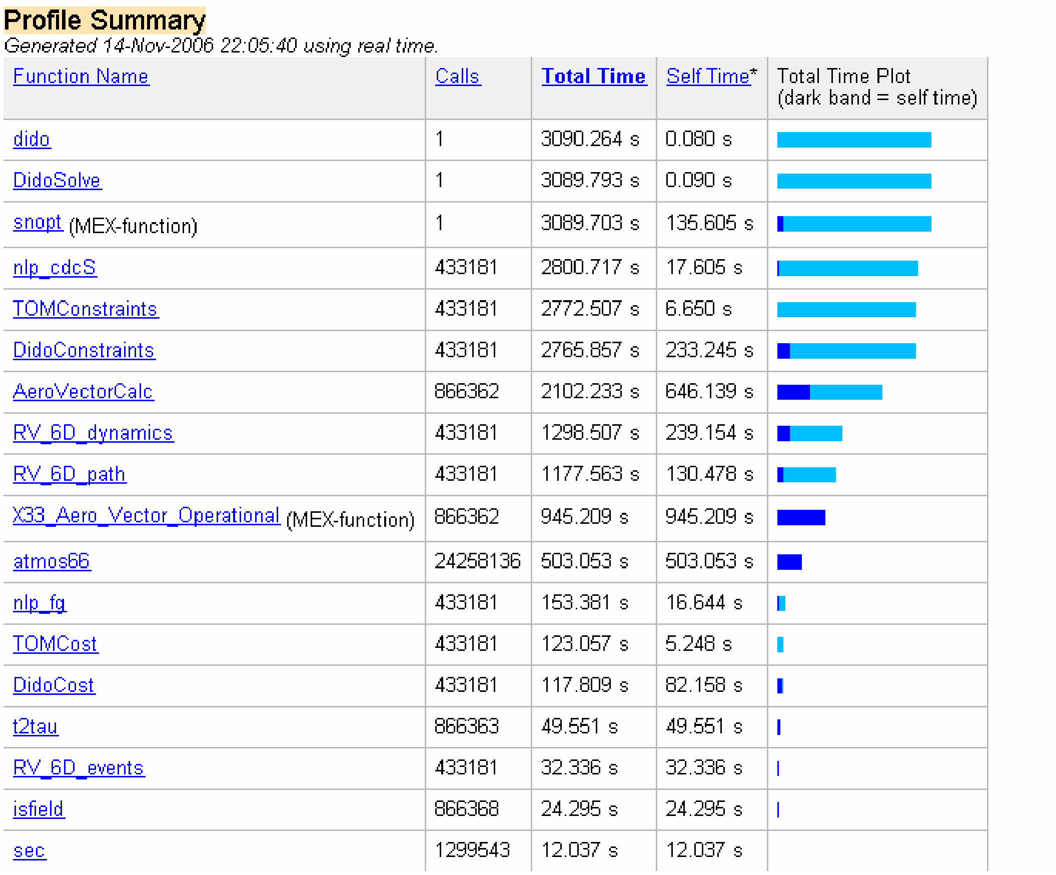


Figure 6.20 Max DR Yawing Moment History.

As shown in Figure 6.12 and Figure 6.13, the max-range cost for this solution is approximately 668,500 ft (110 nm) with a total flight time of about 863 sec. The sideslip history in Figure 6.8 shows rapid oscillations in the beginning and during the second half of the trajectory. For a maximum downrange trajectory, it is expected that the sideslip is zero. This nonzero sideslip may be a sign of this variable's sensitivity in the optimization problem and the underlying physics. Similarly, it is surprising that the roll and heading angles are nonzero as shown in Figure 6.7. Also noteworthy is the large rudder controls as shown in Figure 6.11. Since the rudders should be ineffective for $M > 2.5$, then the solution should be insensitive to these large rudder deflections in the first few-hundred seconds of flight. Figure 6.16 shows large side forces during the approximate same time as the rudder deflections which would indicate that the rudders do have an effect on the solution. This begs the question, why? A more important question is why are the rudders supposed to be ineffective in the first place? Is this restriction based on physics, vehicle limitations, or control effectiveness? Answers to these questions require further research. In general, the resulting forces and moments are consistent with those generated in Chapter V using the 6-DOF PLP allocation for the tracking simulation.

The relatively long computation time of this solution, on average, ranged from 798 sec to 900 sec for the first 14-node solution and then from 3090 sec to 8377 sec (2hr:19min:37sec) for the bootstrapped 28-node solution. Reviewing the MATLAB generated "Profile" revealed that about 52% of this CPU-time was spent performing aerodynamic calculations as indicated by the snapshot in Figure 6.21. Again, it is important to keep in mind that this solution was generated using unoptimized code, running on legacy MATLAB software in the Windows environment with the overhead of other applications using valuable CPU processor speed and memory.



Profile Summary
Generated 14-Nov-2006 22:05:40 using real time.

Function Name	Calls	**Total Time**	Self Time*	Total Time Plot (dark band = self time)
dido	1	3090.264 s	0.080 s	
DidoSolve	1	3089.793 s	0.090 s	
snopt (MEX-function)	1	3089.703 s	135.605 s	
nlp_cdcS	433181	2800.717 s	17.605 s	
TOMConstraints	433181	2772.507 s	6.650 s	
DidoConstraints	433181	2765.857 s	233.245 s	
AeroVectorCalc	866362	2102.233 s	646.139 s	
RV_6D_dynamics	433181	1298.507 s	239.154 s	
RV_6D_path	433181	1177.563 s	130.478 s	
X33_Aero_Vector_Operational (MEX-function)	866362	945.209 s	945.209 s	
atmos66	24258136	503.053 s	503.053 s	
nlp_fg	433181	153.381 s	16.644 s	
TOMCost	433181	123.057 s	5.248 s	
DidoCost	433181	117.809 s	82.158 s	
t2tau	866363	49.551 s	49.551 s	
RV_6D_events	433181	32.336 s	32.336 s	
isfield	866368	24.295 s	24.295 s	
sec	1299543	12.037 s	12.037 s	

Figure 6.21 Snapshot from MATLAB "Profile Summary".

Note that according to MATLAB,

> Self time is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

3. Problem HF1: Homotopy Flat-Earth Model #1

a. Problem Formulation

This model is primarily used for troubleshooting or as a stepping-stone to following a "homotopy" path - a simpler model with a logical path to a more complex problem. Here, the forces and moments are modeled as the controls. A control allocation problem usually solves for control effectors that produce pitch, yaw, and roll moments and is not directly concerned with force generation. In other words, the control allocation is attempting to control the wing-body attitude to generate the desired forces. This problem formulation differs in that both the moments and the forces are designated as the controls. The purpose of this formulation is to validate the standard equations used in the flat-earth model (Problem SF) and to help distinguish between the problem formulation, specifically the EoM, and the aerodynamic data as to the root cause of any problems. The control vector and the corresponding control variable bounds, respectively, are defined as

$$\mathbf{u} = [X, Y, Z, L, M, N] \in \mathbb{R}^6 \tag{5.50}$$

$$\begin{array}{ll} -200000 \le X \le 0 & \text{(lb)} \\ -1000 \le Y \le 1000 & \text{(lb)} \\ -300000 \le Z \le 100000 & \text{(lb)} \\ -1000 \le L \le 1000 & \text{(ft-lb)} \\ -20000 \le M \le 20000 & \text{(ft-lb)} \\ -1000 \le N \le 1000 & \text{(ft-lb)} \end{array} \tag{5.51}$$

where the boundary limits for each control variable were selected based on the approximate minimum and maximum values observed in the 6-DOF simulation from the previous section.

b. Results and Analysis

These results were generated by bootstrapping from 20 to 80 nodes. Recall that this model did not rely on the aerodynamic model; therefore, substantially faster computational times facilitate the use of higher node solutions. The following figures present some of the results from this solution.

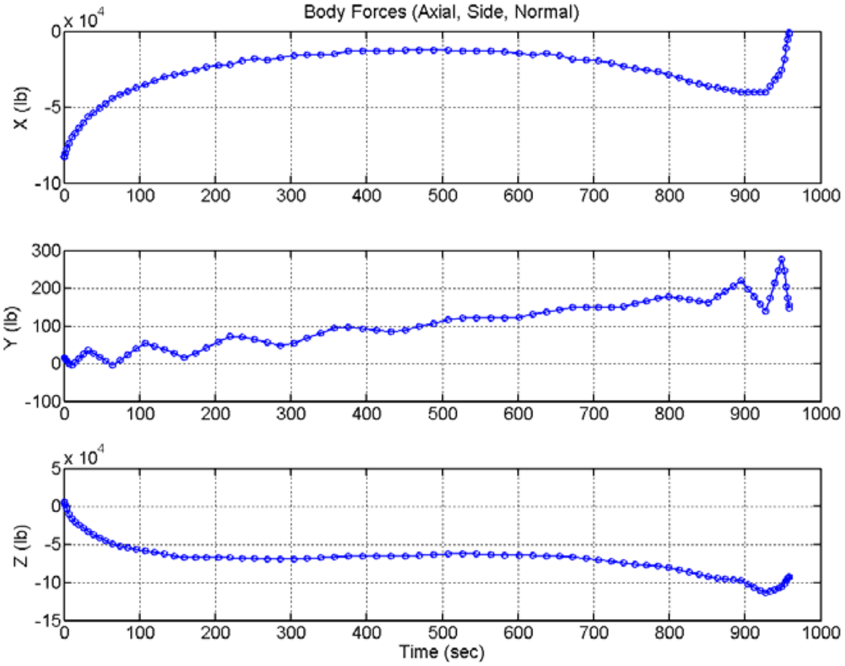


Figure 6.22 Max DR Force Controls.

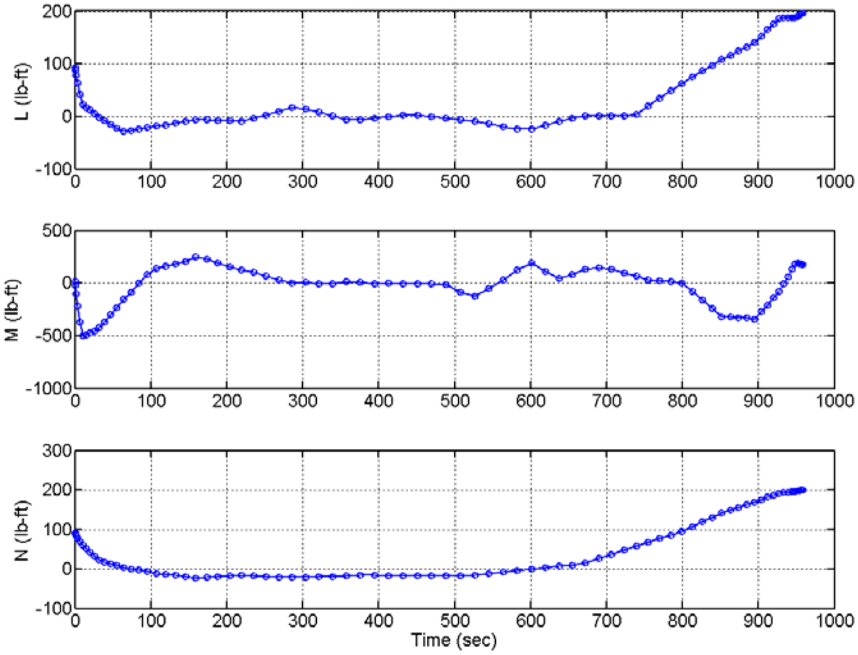


Figure 6.23 Max DR Moment Controls.

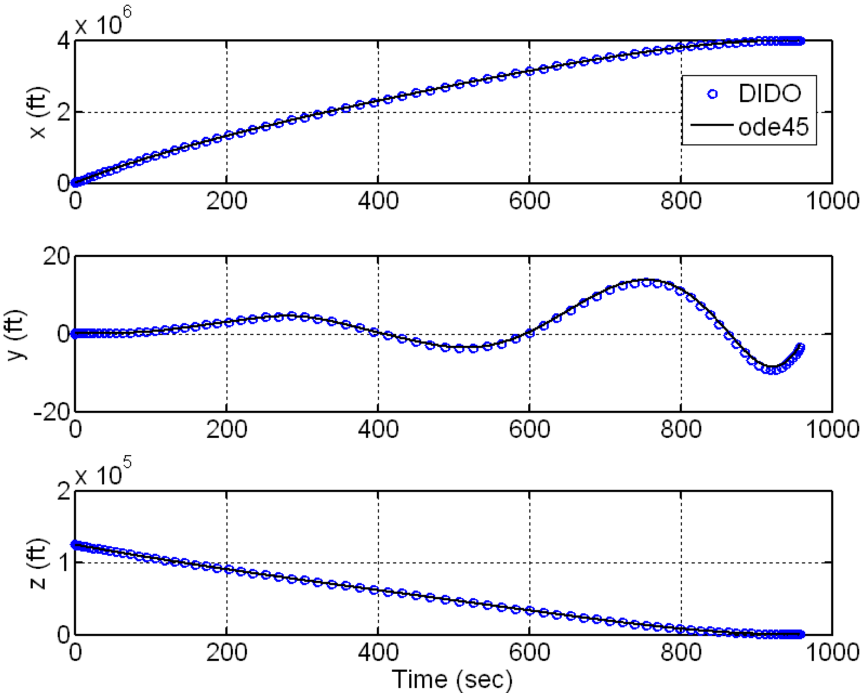


Figure 6.24 Max DR Cartesian Position.

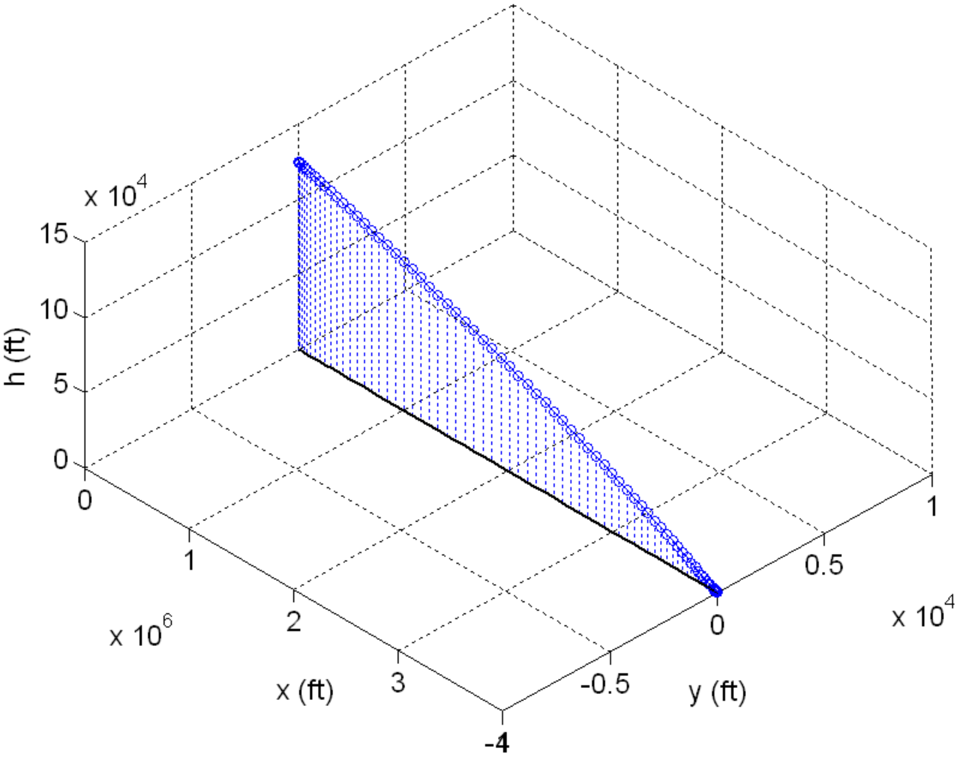


Figure 6.25 Max DR 3D Flight Trajectory Profile.

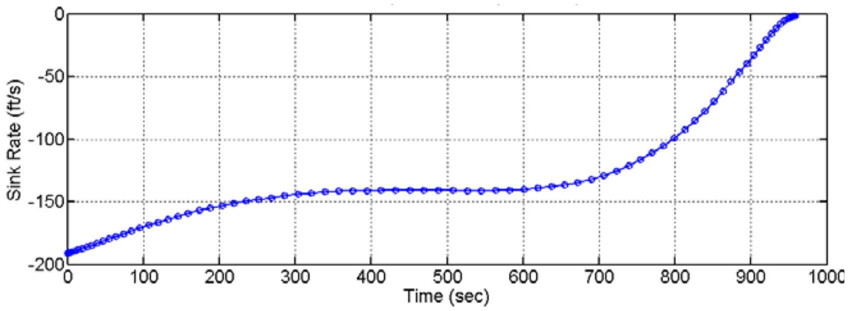


Figure 6.26 Max DR Vertical Sink Rate Profile.

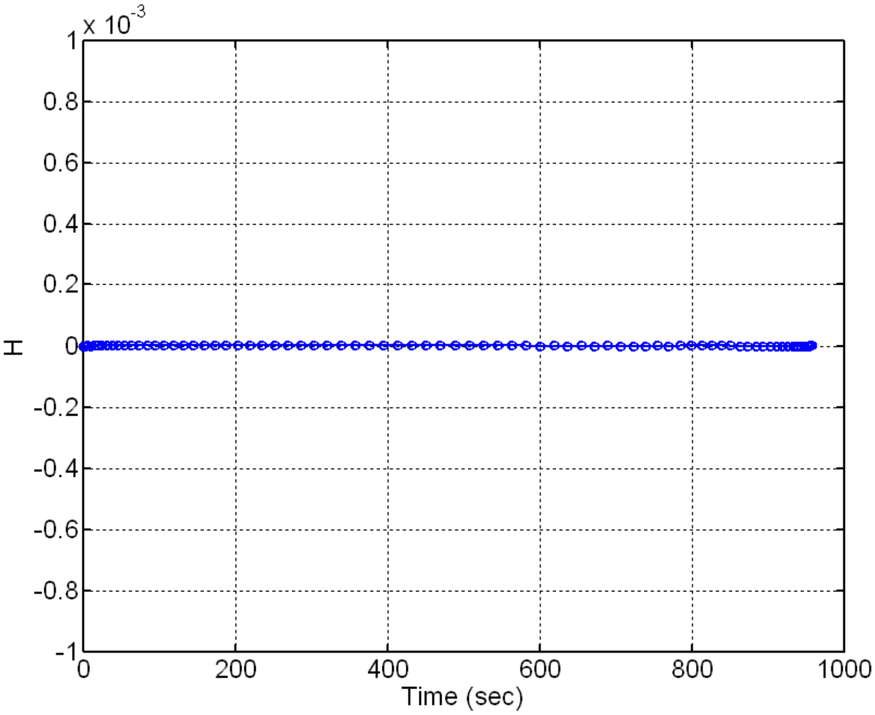


Figure 6.27 Max DR Hamiltonian Function.

The results from this model are not physically realistic in the sense that the homotopy controls, the forces in this case, are not limited by atmospheric nor aerodynamic effects. As such, the maximum downrange is essentially unlimited as indicated by the excessively large value of 4,000,000 ft (658 nm) in Figure 6.24 and Figure 6.25. The importance of these results, although not realistic, is to verify that the computational time for the aerodynamic database and the corresponding dynamics are independent of any other modeling issues. For this run, the 80-node solution was generated in approximately 69 sec, a significant difference compared to the 28-node solution of the previous model that took 3090 sec.

4. Problem HF2: Homotopy Flat-Earth Model #2

a. Problem Formulation

This problem formulation pushes the controls down to the next level, defining them as the aerodynamic force and moment *coefficients*, as opposed to just the forces and moments as in the previous case. The control vector and the corresponding control variable bounds, respectively, are succinctly defined as

$$\mathbf{u} = [C_A, C_Y, C_N, C_l, C_m, C_n] \in \mathbb{R}^6 \tag{5.52}$$

$$\begin{aligned} -0.040 &\le C_A \le 0.320 \\ -0.170 &\le C_Y \le 0.170 \\ -0.360 &\le C_N \le 1.630 \\ -0.073 &\le C_l \le 0.073 \\ -0.040 &\le C_m \le 0.050 \\ -0.030 &\le C_n \le 0.030 \end{aligned} \tag{5.53}$$

According to Eqs. (3.82)-(3-84), the corresponding upper and lower bounds on the lift, drag, and side force coefficients should coincide with the maximum and minimum values given in Eq. (5.53). This is verified by solving subsequent minimization/maximization problems. For example, the maximum lift coefficient is found by solving

$$\begin{cases} Max \;\; C_L = -C_A \sin\alpha + C_N \cos\alpha \\ s.t. \quad -10 \le \alpha \le 50 \\ \qquad\;\; -0.040 \le C_A \le 0.320 \\ \qquad\;\; -0.360 \le C_N \le 1.630 \end{cases} \tag{5.54}$$

As expected, the solution for this problem and corresponding problems for drag and side force coefficients results in

$$\begin{aligned} -0.360 &\le C_L \le 1.63 \\ -0.040 &\le C_D \le 0.320 \\ -0.170 &\le C_S \le 0.170 \end{aligned} \tag{5.55}$$

Note that the values used in Eq. (5.53) were taken from the X-33 aerodynamic database without regard to the Mach-alpha space; rather, max/min values were extracted irrespective of the flight condition. As a result of this, the values do not accurately reflect the nominal lift and drag as confirmed from surface plots explicitly depending on Mach and alpha. According to the lift and drag plots in Chapter IV, the data indicates the max/min lift and drag coefficients are

$$\begin{aligned} C_{L_{\min}} &= -0.2370; \quad C_{L_{\max}} = 0.8711 \\ C_{D_{\min}} &= \ \ 0.0953; \quad C_{D_{\max}} = 0.8744 \end{aligned} \tag{5.56}$$

Given this discrepancy, it is more appropriate to add the following path constraints

$$\begin{aligned} & C_{L_{\min}} \le -C_A \sin\alpha + C_N \cos\alpha \le C_{L_{\max}} \\ & C_{D_{\min}} \le C_A \cos\alpha \cos\beta - C_Y \sin\beta + C_N \sin\alpha \cos\beta \le C_{D_{\max}} \\ & C_{S_{\min}} \le C_A \cos\alpha \sin\beta - C_Y \cos\beta - C_N \sin\alpha \sin\beta \le C_{S_{\max}} \end{aligned} \tag{5.57}$$

b. Results and Analysis

Similar to the previous case, these results were generated by bootstrapping from 20 to 80 nodes. Taking approximately the same computational time, these results did account for density-altitude effects since the controls were the aerodynamic coefficients as opposed to the forces in the previous section.

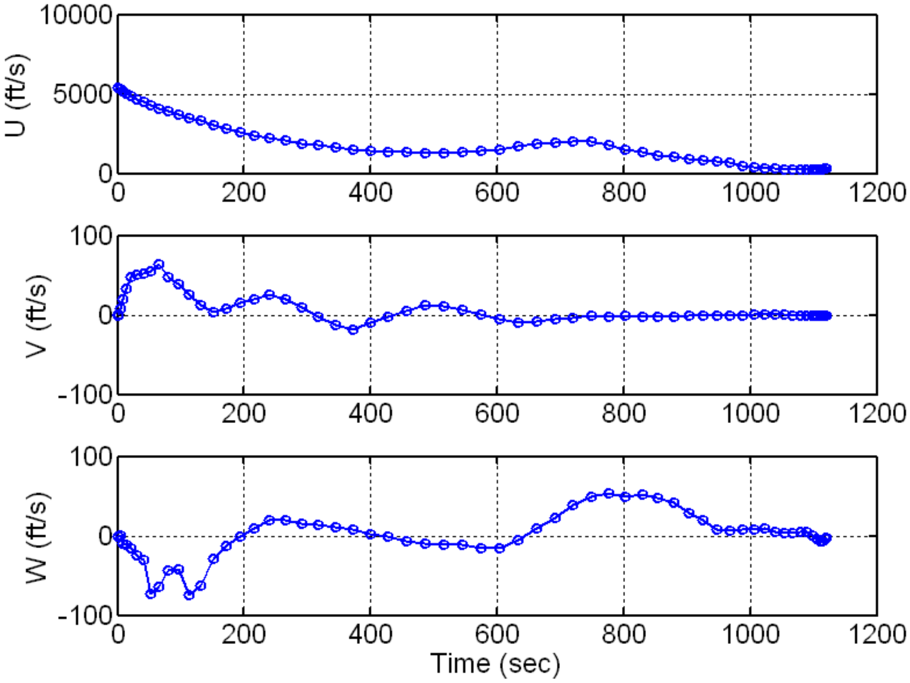


Figure 6.28 Max DR Linear Body-Relative Velocities.

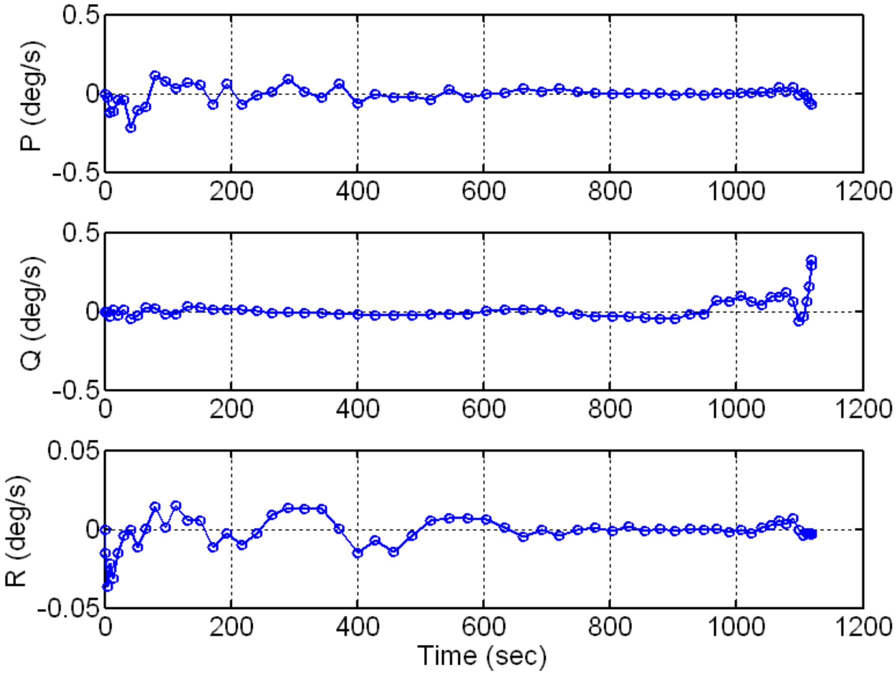


Figure 6.29 Max DR Angular Body-Rates.

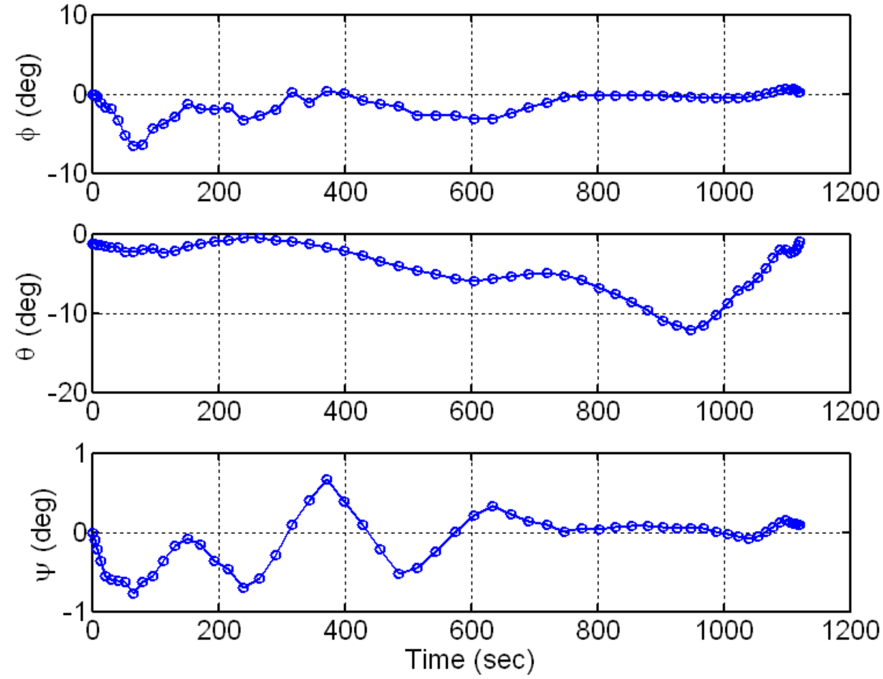


Figure 6.30 Max DR Euler Angles.

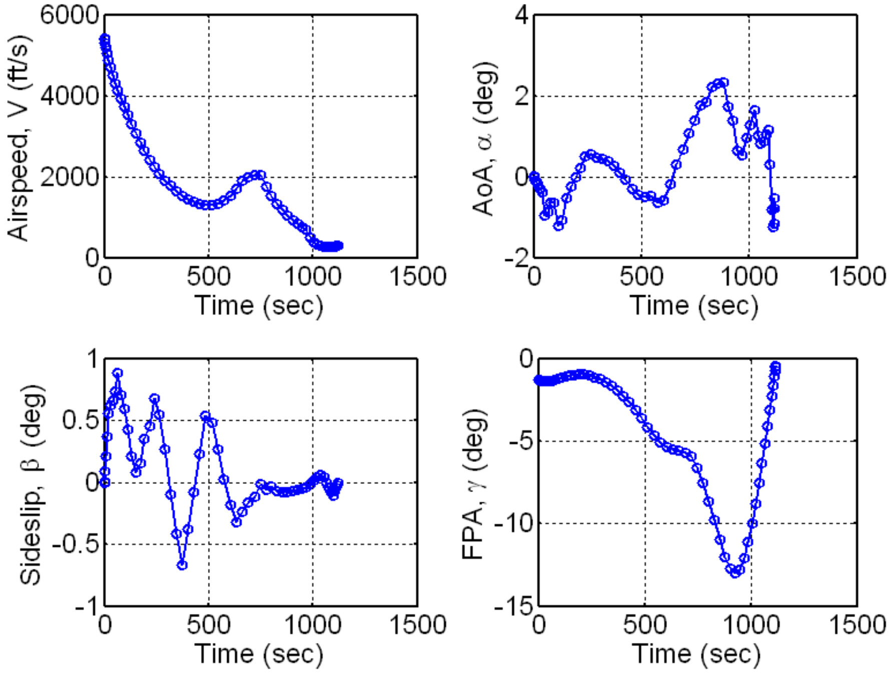


Figure 6.31 Max DR Airspeed and Wind-Relative Angles.

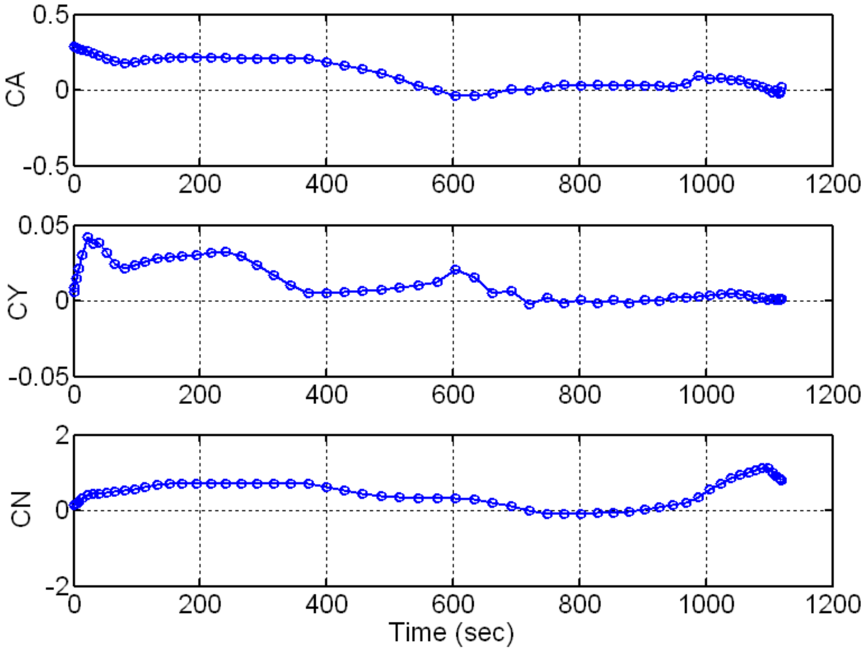


Figure 6.32 Max DR Force Coefficient Controls.

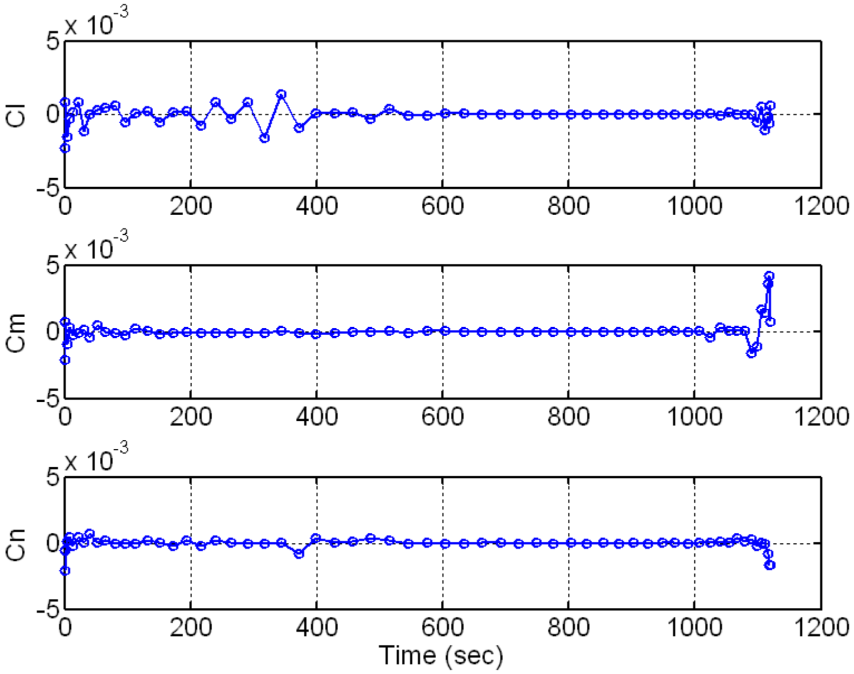


Figure 6.33 Max DR Moment Coefficient Controls.

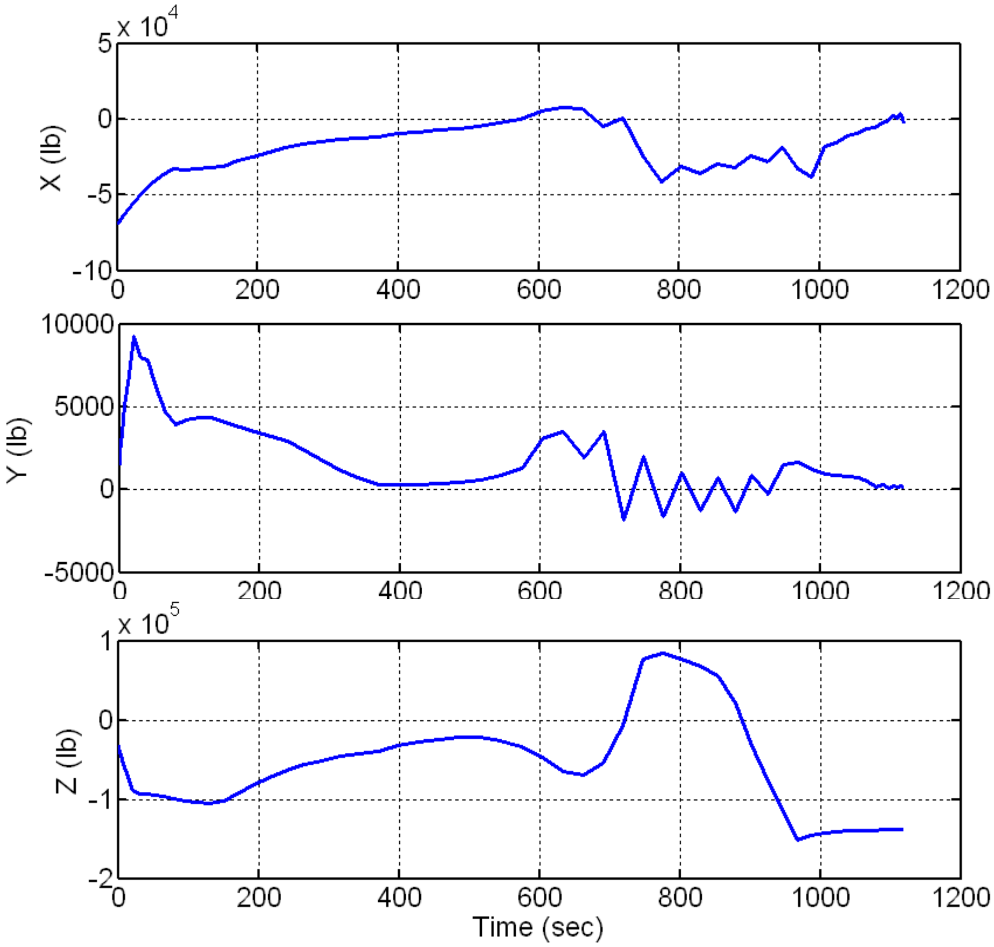


Figure 6.34 Max DR Force History (Axial, Side, Normal).

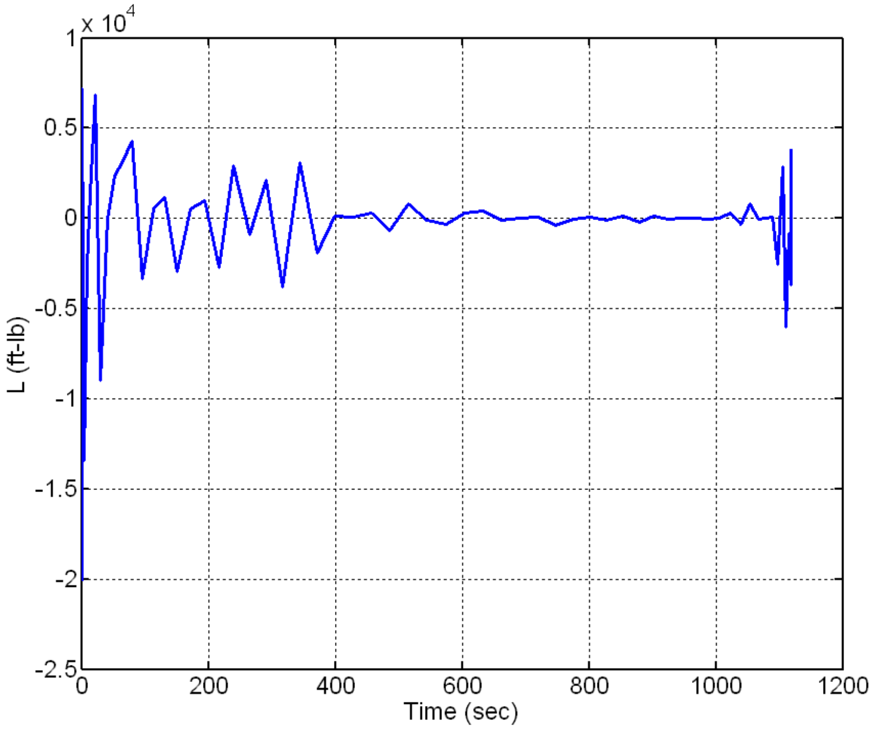


Figure 6.35 Max DR Rolling Moment History.

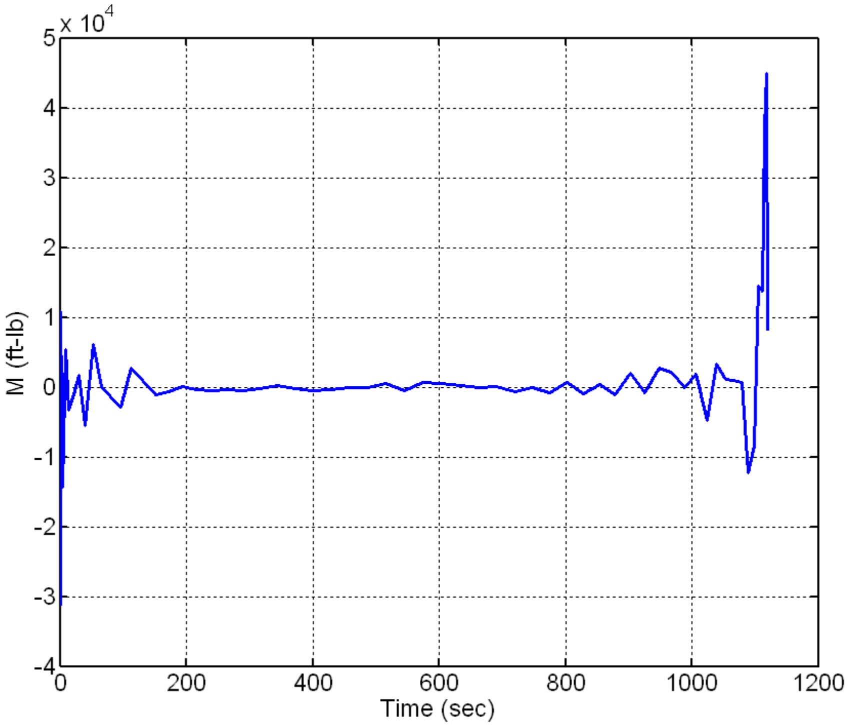


Figure 6.36 Max DR Pitching Moment History.

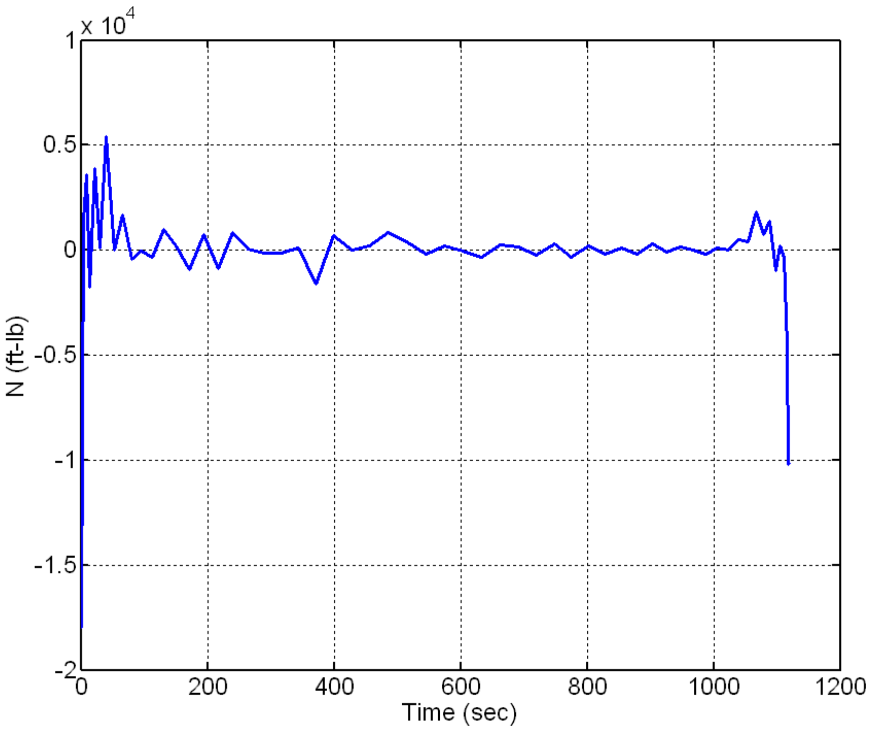


Figure 6.37   Max DR Yawing Moment History.

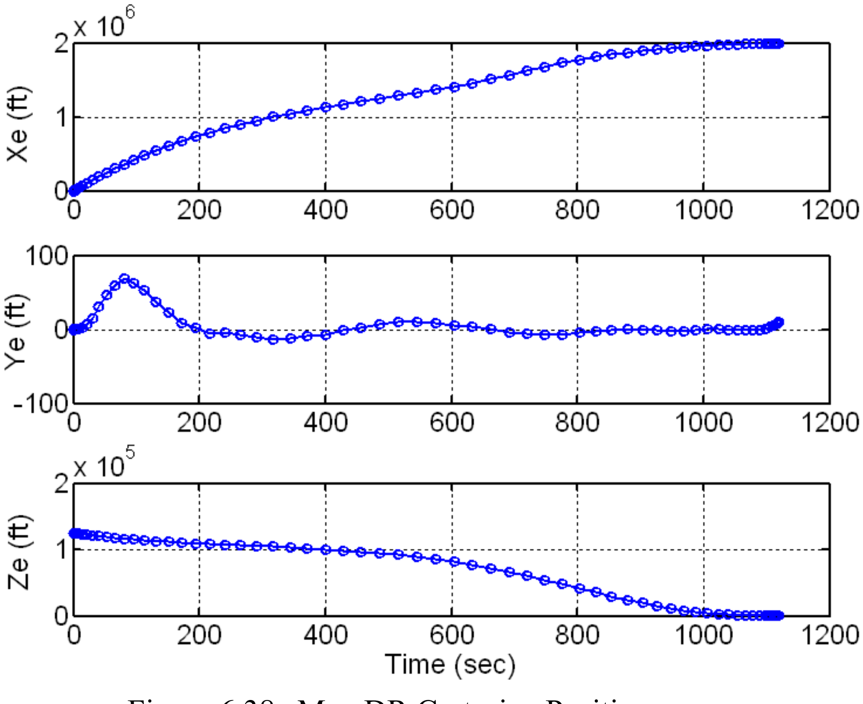


Figure 6.38   Max DR Cartesian Position.

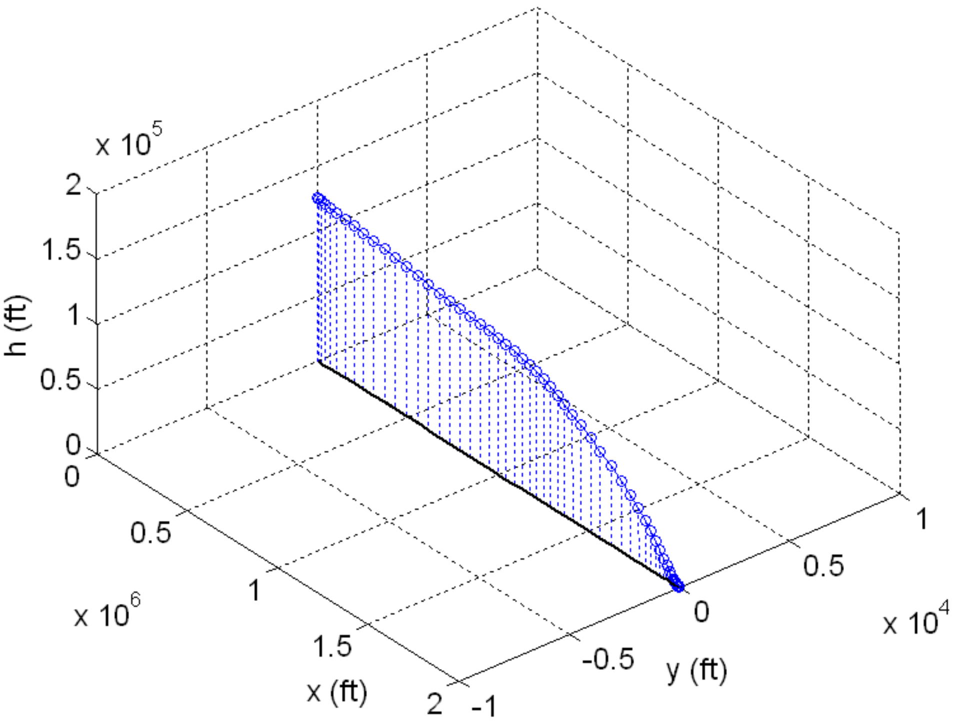


Figure 6.39 Max DR 3D Flight Trajectory Profile.

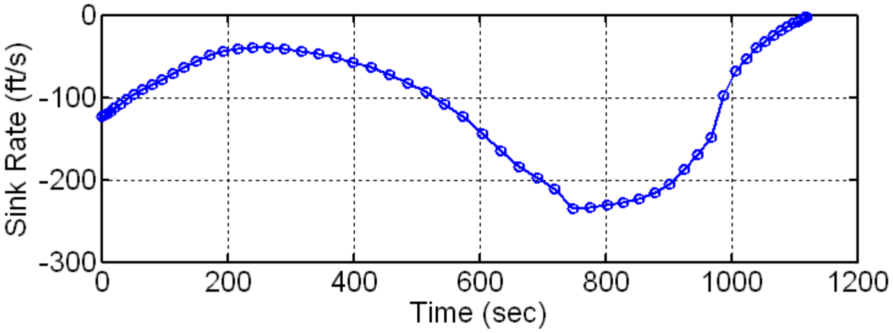


Figure 6.40 Max DR Vertical Sink Rate Profile.

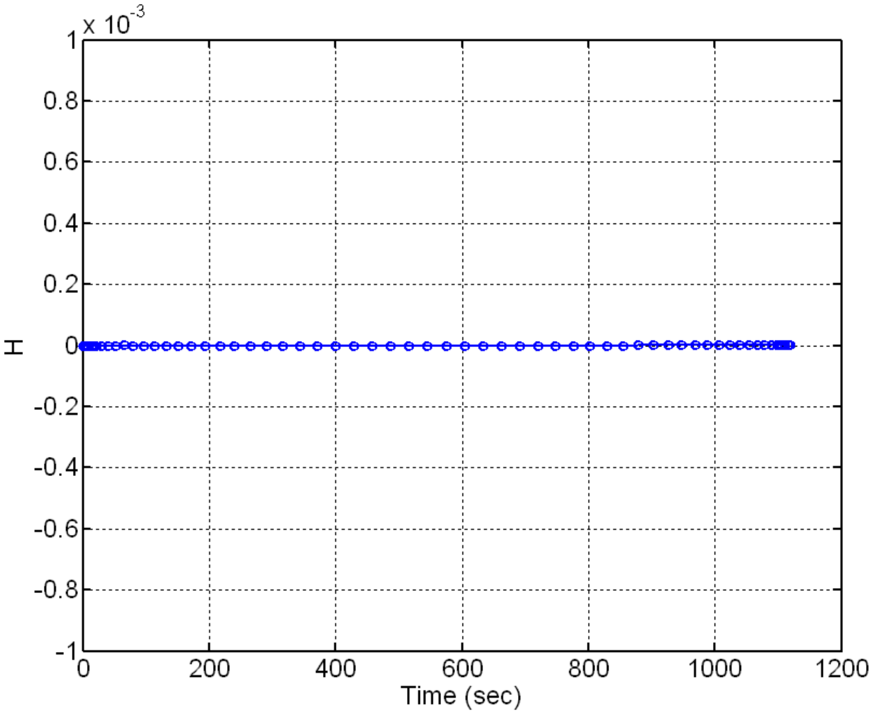


Figure 6.41 Max DR Hamiltonian Function.

As expected, Figure 6.32 shows that the solution tends to minimize the axial force coefficient and the side force coefficient, while maximizing the normal force coefficient. Similar to the previous solution and for the same reason, these results also give unrealistic maximum downrange performance as indicated in Figure 6.38 and Figure 6.39 with a downrange of 2,000,000 ft (329 nm).

5. Problem SS: Standard Spherical-Earth Model

a. Problem Formulation

As described in Chapter III, this model employs a North-East-Down (NED) coordinate system with the equations expressed in a mixed wind-body reference frame. That is, the translational equations are with respect to the velocity or wind relative reference frame and the rotational equations are with respect to the body-fixed reference frame. As such, an important aspect of this model is that the translational and rotational equations are coupled strictly by the aerodynamics via the flight angles, α and β. Since these angles are not explicit state variables as in the standard equations, Problem SF, they must be calculated in order to obtain the aerodynamic coefficients. This model uses the approximate flight angle relationships as discussed in Chapter III, in addition to imposing the requirement for coordinated turns, i.e., zero sideslip angles. This will be useful for comparing to the 3-DOF solutions since it assumes zero sideslip as well. In addition, this model uses a spherical, rotating earth with an inverse-squared gravitational field.

For this problem, the state and control vectors are defined as

$$\begin{aligned} \underline{x} &= [h, \mu, \lambda, V, \gamma, \xi, p, q, r, \phi, \theta, \psi]^T \in \mathbb{R}^{12} \\ \underline{u} &= \underline{\delta} = [\delta_{RE_{in}}, \delta_{RE_{out}}, \delta_{LE_{in}}, \delta_{LE_{out}}, \delta_{RF}, \delta_{LF}, \delta_{RR}, \delta_{LR}]^T \in \mathbb{R}^8 \\ &\equiv [u_1, ..., u_8] \end{aligned} \tag{5.58}$$

Again, the control vector assumes there are no command delays. To maximize the range, the cost function is subject to the following constraints.

Dynamic Constraints, $\dot{x} = f(x(t), u(t), t)$:

$$\dot{\underline{x}} = [\dot{h}, \dot{\mu}, \dot{\lambda}, \dot{V}, \dot{\gamma}, \dot{\xi}, \dot{p}, \dot{q}, \dot{r}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T = f(x, u, t) \tag{5.59}$$

where $f(x,u,t)$ is the right-hand-side (RHS) of the dynamic EoM specified in Figure 3.14 (see Chapter III).

Path Constraints, $h^L \leq h(x(t), u(t), t) \leq h^U$:

$$\begin{bmatrix} -2.5\,\text{g's} \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} n_z(h, V, \alpha) \\ \overline{q}(h, V) \\ Q(h, V) \end{bmatrix} \leq \begin{bmatrix} 2.5\,\text{g's} \\ 300\,\text{lb/ft}^2 \\ 60\,\text{BTU/ft-s} \end{bmatrix} \tag{5.60}$$

Initial and Final Endpoint Conditions, $e^L \leq e(x(t_0), x(t_f), t_0, t_f) \leq e^U$:

$$\begin{array}{llll} \text{h}(\text{t}_0) = 125{,}000 \text{ ft} & \text{V}(\text{t}_0) = 5413 \text{ ft/s} & \text{p}(t_0) = 0\,\text{deg/s} & \phi(t_0) = 0\,\text{deg} \\ \mu(\text{t}_0) = 0 \text{ deg} & \gamma(t_0) = \text{-}1.3 \text{ deg} & \text{q}(t_0) = 0\,\text{deg/s} & \theta(t_0) = 18 \text{ deg} \\ \lambda(\text{t}_0) = 0 \text{ deg} & \xi(t_0) = 0 \text{ deg} & \text{r}(t_0) = 0\,\text{deg/s} & \psi(t_0) = 0 \text{ deg} \end{array} \tag{5.61}$$

$$\begin{aligned} h(t_f) &= 500\,\text{ft} \\ V(t_f) &= 335.13\,\text{ft/s} \\ \gamma(t_f) &= -3\,\text{deg} \end{aligned} \tag{5.62}$$

Note that the final altitude and final FPA correspond to a final sink rate of $V\sin\gamma = -17.54\,\text{ft/s}$.

Initial and Final Guesses:

$$\begin{array}{ll}
h_0 = 125{,}000 \text{ ft} & h_\mathrm{f} = 500 \text{ ft} \\
\mu_0 = 0 \text{ deg} & \mu_\mathrm{f} = 0 \text{ deg} \\
\lambda_0 = 0\,\text{deg} & \lambda_\mathrm{f} = 0\,\text{deg} \\
\mathrm{V}_0 = 5413 \text{ ft/s} & \mathrm{V}_\mathrm{f} = 335.13 \text{ ft/s} \\
\gamma_0 = \text{-}1.3 \text{ deg} & \gamma_\mathrm{f} = -3 \text{ deg} \\
\xi_0 = 0 \text{ deg} & \xi_\mathrm{f} = 0 \text{ deg} \\
\mathrm{p}_0 = 0\,\text{deg/s} & \mathrm{p}_\mathrm{f} = 0\,\text{deg/s} \\
\mathrm{q}_0 = 0\,\text{deg/s} & \mathrm{q}_\mathrm{f} = 0\,\text{deg/s} \\
\mathrm{r}_0 = 0\,\text{deg/s} & \mathrm{r}_\mathrm{f} = 0\,\text{deg/s} \\
\phi_0 = 0\,\text{deg} & \phi_\mathrm{f} = 0\,\text{deg} \\
\theta_0 = 18 \text{ deg} & \theta_\mathrm{f} = 0 \text{ deg} \\
\psi_0 = 0 \text{ deg} & \psi_\mathrm{f} = 0 \text{ deg}
\end{array} \tag{5.63}$$

$$\begin{array}{ll}
\delta_{\mathrm{REi}_0} = \text{-}0.8 \text{ deg} & \delta_{\mathrm{REi}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{REo}_0} = \text{-}0.8\,\text{deg} & \delta_{\mathrm{REo}_\mathrm{f}} = 0\,\text{deg} \\
\delta_{\mathrm{LEi}_0} = \text{-}0.8\,\text{deg} & \delta_{\mathrm{LEi}_\mathrm{f}} = 0\,\text{deg} \\
\delta_{\mathrm{LEo}_0} = \text{-}0.8 \text{ deg} & \delta_{\mathrm{LEo}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{RF}_0} = 7.9 \text{ deg} & \delta_{\mathrm{RF}_\mathrm{f}} = 0 \text{ deg} \\
\delta_{\mathrm{LF}_0} = 7.9\,\text{deg} & \delta_{\mathrm{LF}_\mathrm{f}} = 0\,\text{deg} \\
\delta_{\mathrm{RR}_0} = 0\,\text{deg} & \delta_{\mathrm{RR}_\mathrm{f}} = 0\,\text{deg} \\
\delta_{\mathrm{LR}_0} = 0\,\text{deg} & \delta_{\mathrm{LR}_\mathrm{f}} = 0\,\text{deg}
\end{array} \tag{5.64}$$

State Variable Constraints, $x^L \le x(t) \le x^U$:

$$\begin{array}{ll}
0 \text{ ft} \le h \le 4e5 \text{ ft} & -30\,\text{deg/s} \le p \le 30\,\text{deg/s} \\
0\,\text{deg} \le \mu \le 10 \text{ deg} & -30\,\text{deg/s} \le q \le 30\,\text{deg/s} \\
-4 \text{ deg} \le \lambda \le 4\,\text{deg} & -30\,\text{deg/s} \le r \le 30\,\text{deg/s} \\
1.0\,\text{ft/s} \le V \le 2 * V(t_0)\,\text{ft/s} & -90\,\text{deg} \le \phi \le 90\,\text{deg} \\
-80\,\text{deg} \le \gamma \le 80\,\text{deg} & -89\,\text{deg} \le \theta \le 89\,\text{deg} \\
-180\,\text{deg} \le \xi \le 180\,\text{deg} & -180\,\text{deg} \le \psi \le 180\,\text{deg}
\end{array} \tag{5.65}$$

and Control Variable Constraints, $u^L \le u(t) \le u^U$:

$$\begin{aligned} -30\,\deg \le \delta_i \le 30\,\deg & \qquad i = 1,2,3,4 \\ -15\,\deg \le \delta_j \le 26\,\deg & \qquad j = 5,6 \\ -60\,\deg \le \delta_7 \le 30\,\deg & \\ -30\,\deg \le \delta_8 \le 60\,\deg & \end{aligned} \tag{5.66}$$

In addition to the regular path constraints of heating rate, dynamic pressure, and normal acceleration, the approximate flight angle relationships,

$$\begin{aligned} \alpha &= \theta - \gamma \\ \beta &= \xi - \psi \end{aligned} \tag{5.67}$$

are bounded by implementing the corresponding path constraints

$$\begin{aligned} -10 \le \alpha\left(\theta,\gamma,t\right) \le 50 \\ -1.5 \le \beta\left(\xi,\psi,t\right) \le 1.5 \end{aligned} \tag{5.68}$$

However, since this model assumes coordinated turns by setting $\beta(t) = 0$ for all time, $t \in [t_0, t_f]$, the constraint on sideslip angle is omitted for now.

b. Results and Analysis

The following results were generated by bootstrapping from 14 to 28 nodes which takes on average about 800 sec to 1800 sec compared to the 3090 sec to 8377 sec for the model using the standard EoM (Problem SF).

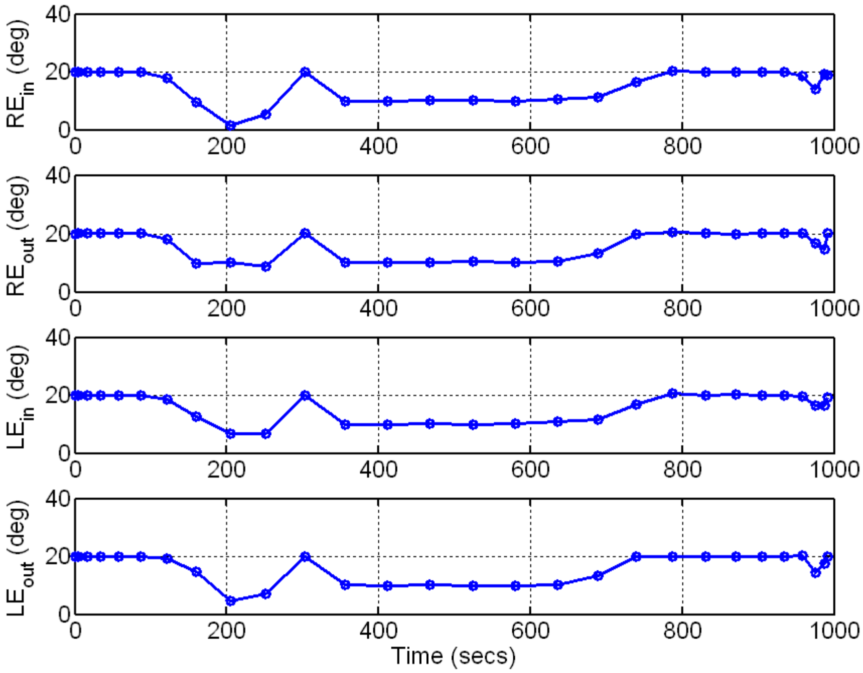


Figure 6.42 Max DR Elevon Controls.

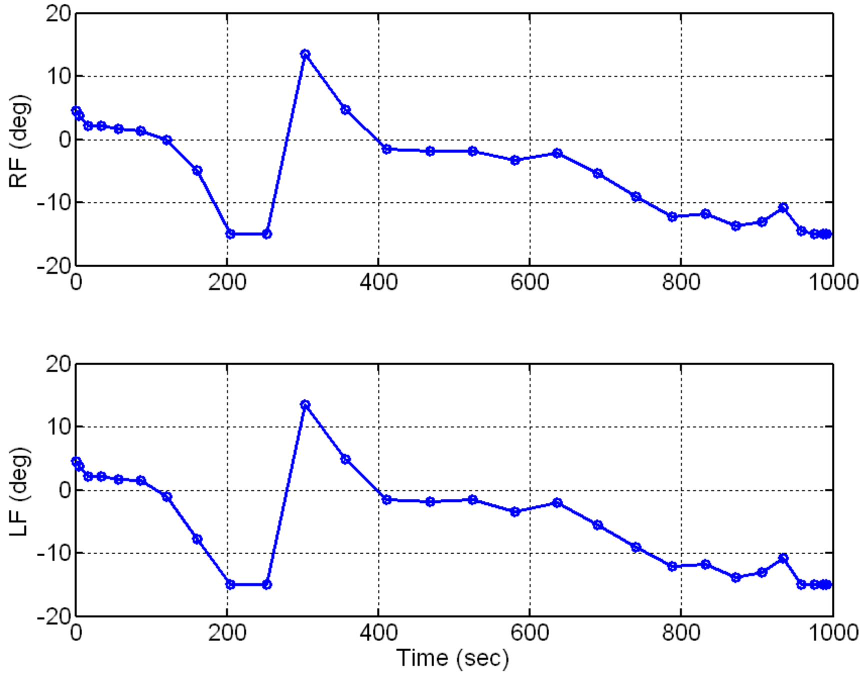


Figure 6.43 Max DR Flap Controls.

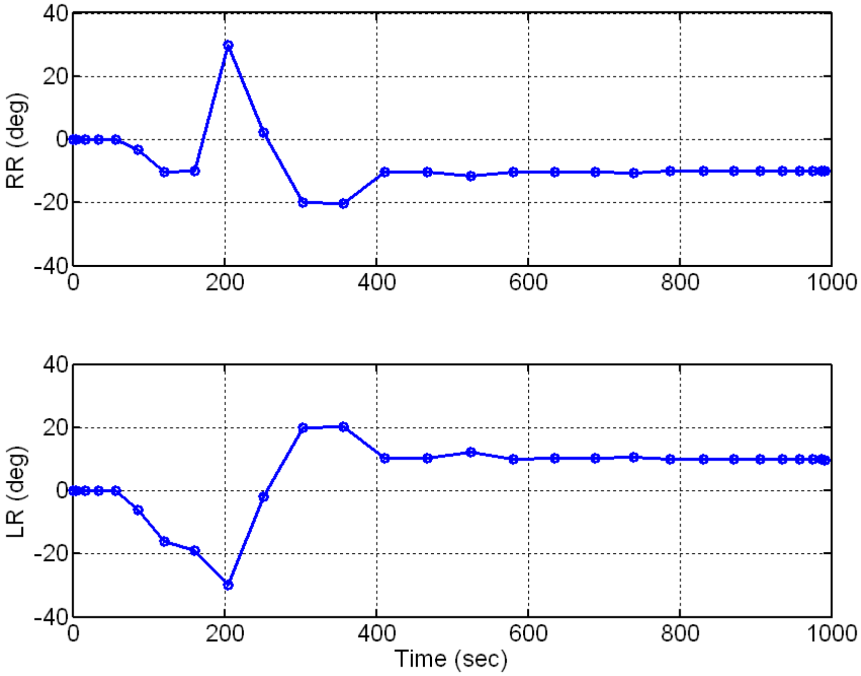


Figure 6.44   Max DR Rudder Controls.

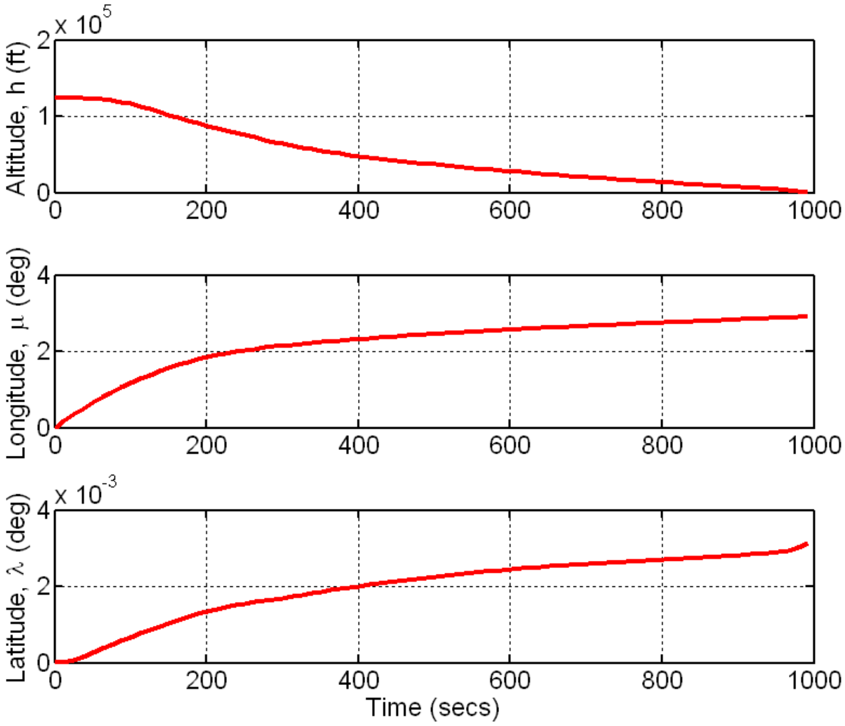


Figure 6.45   Max DR Position Response.

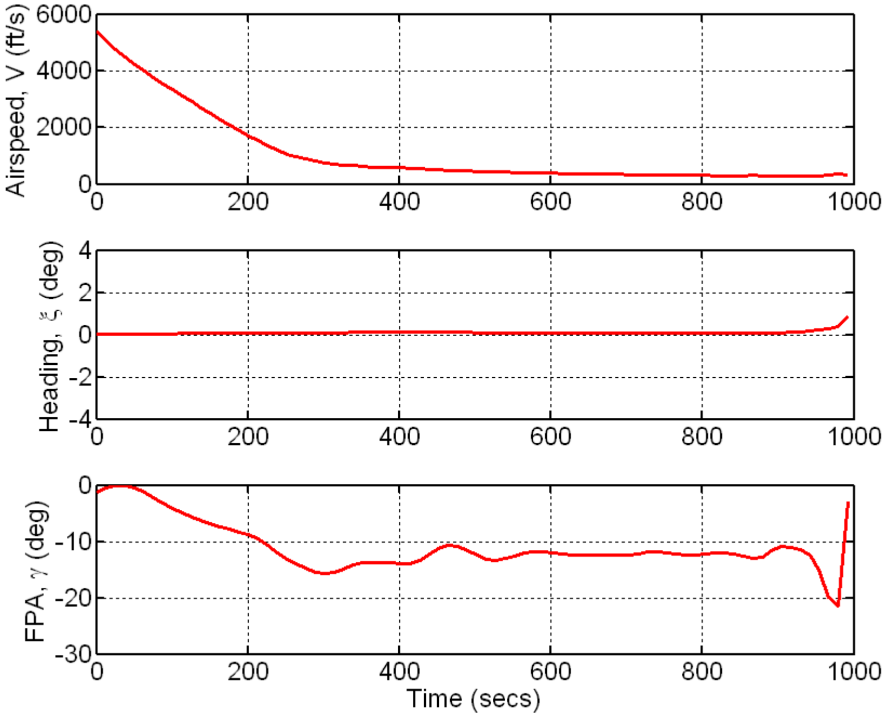


Figure 6.46 Max DR Velocity Response.

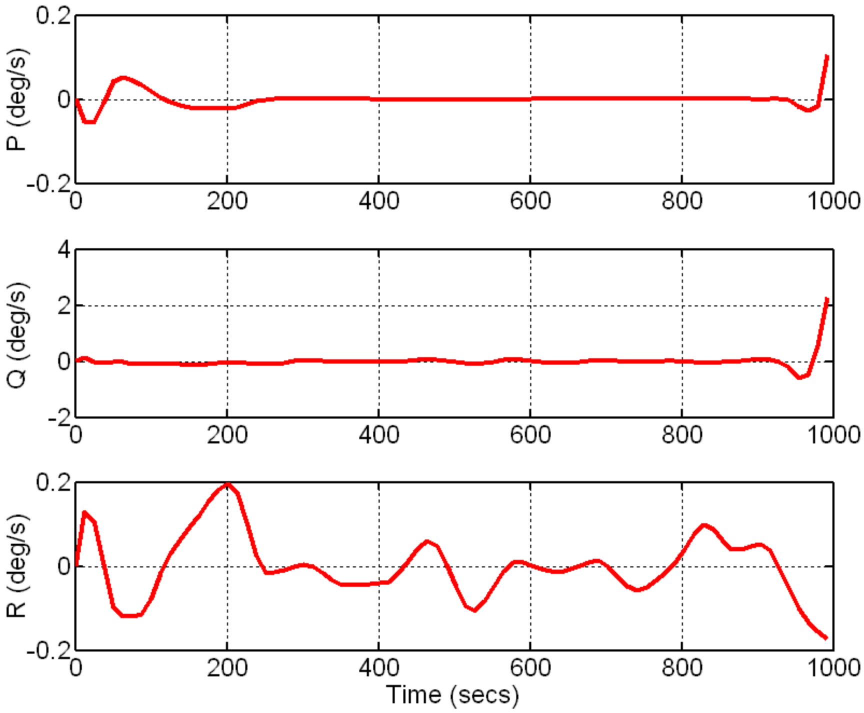


Figure 6.47 Max DR Body Rate Response (degrees).

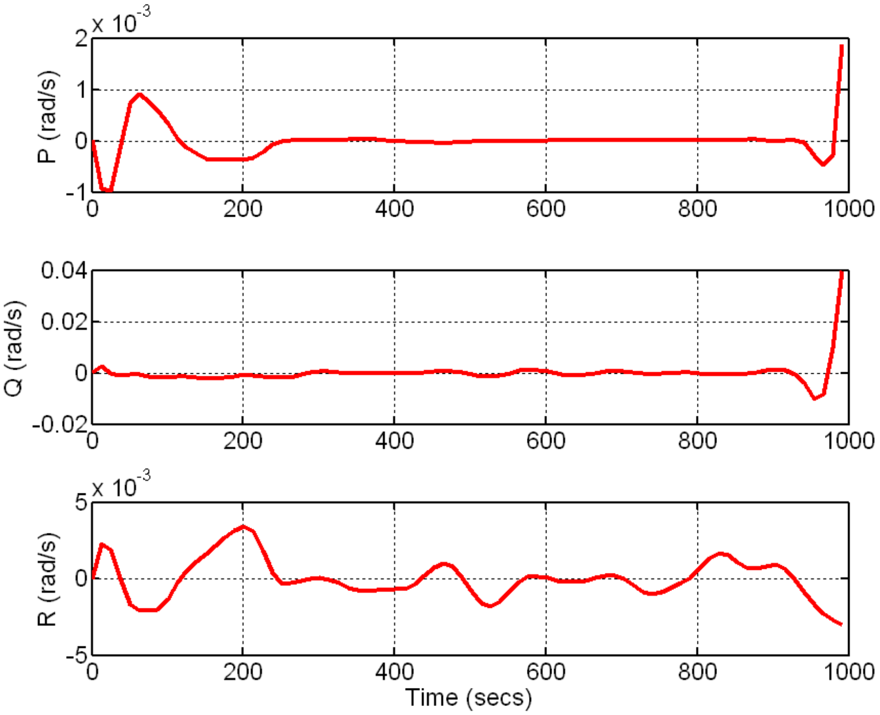


Figure 6.48 Max DR Body Rate Response (Radians).

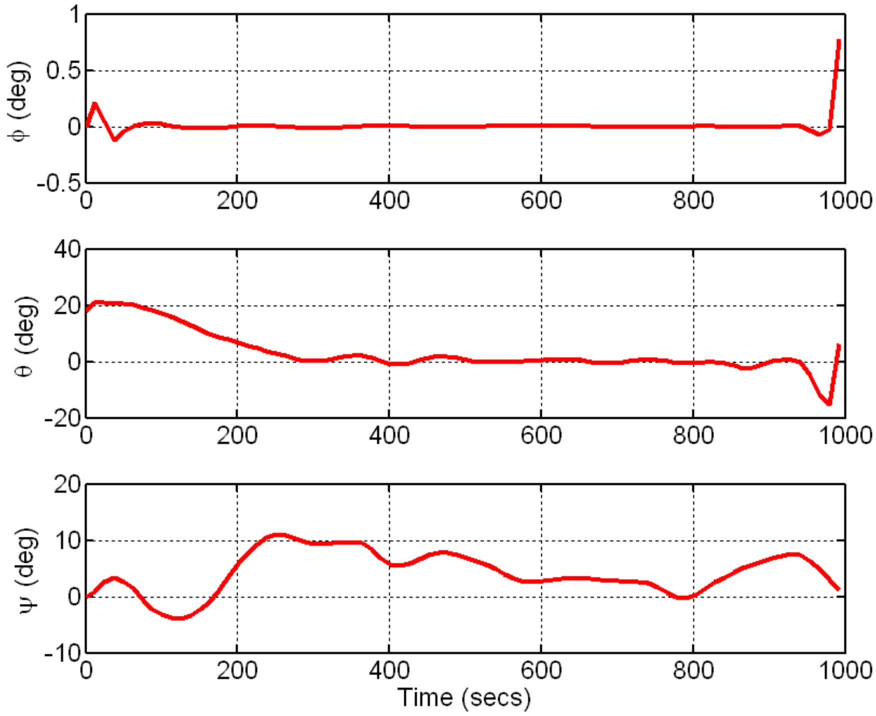


Figure 6.49 Max DR Euler Angle Response.

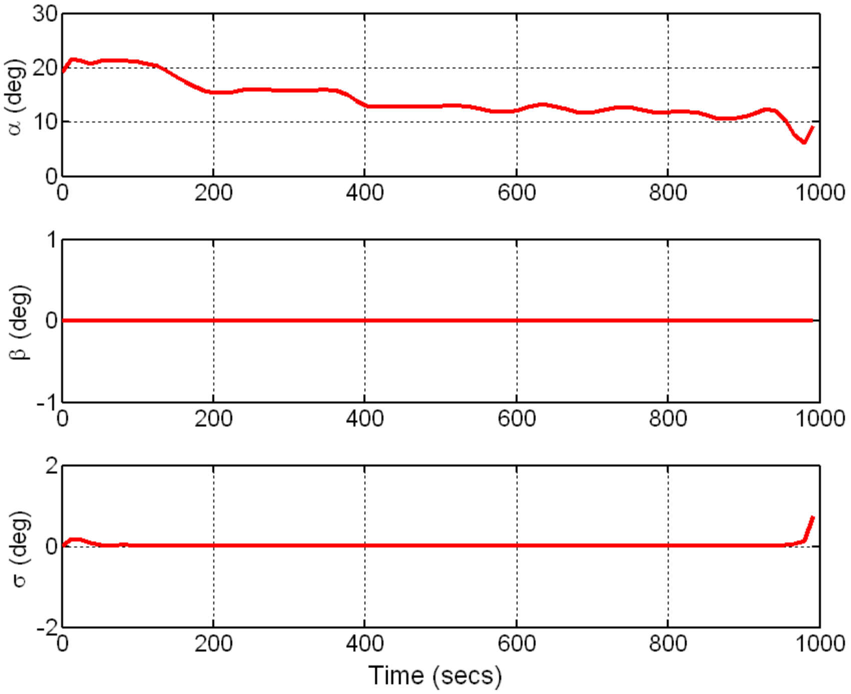


Figure 6.50 Max DR Aero Angle Response.

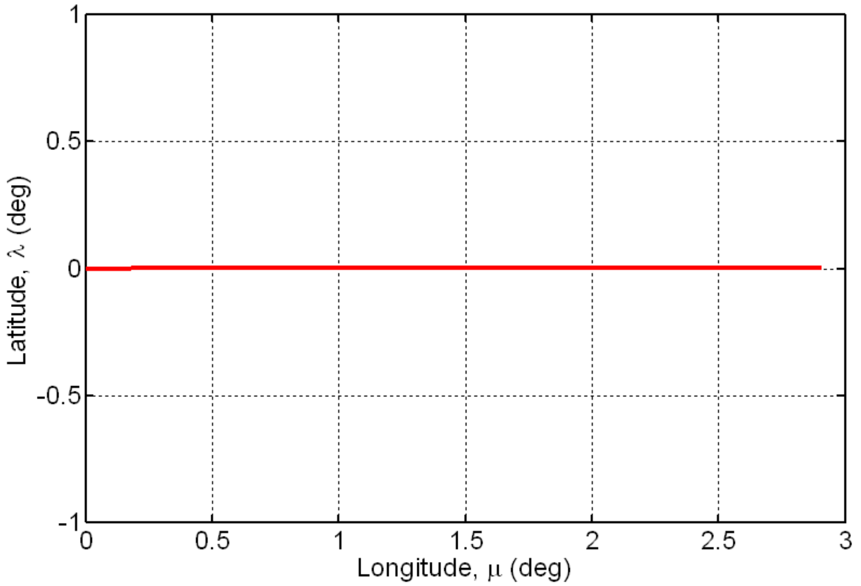


Figure 6.51 Max DR 2D Ground Track.

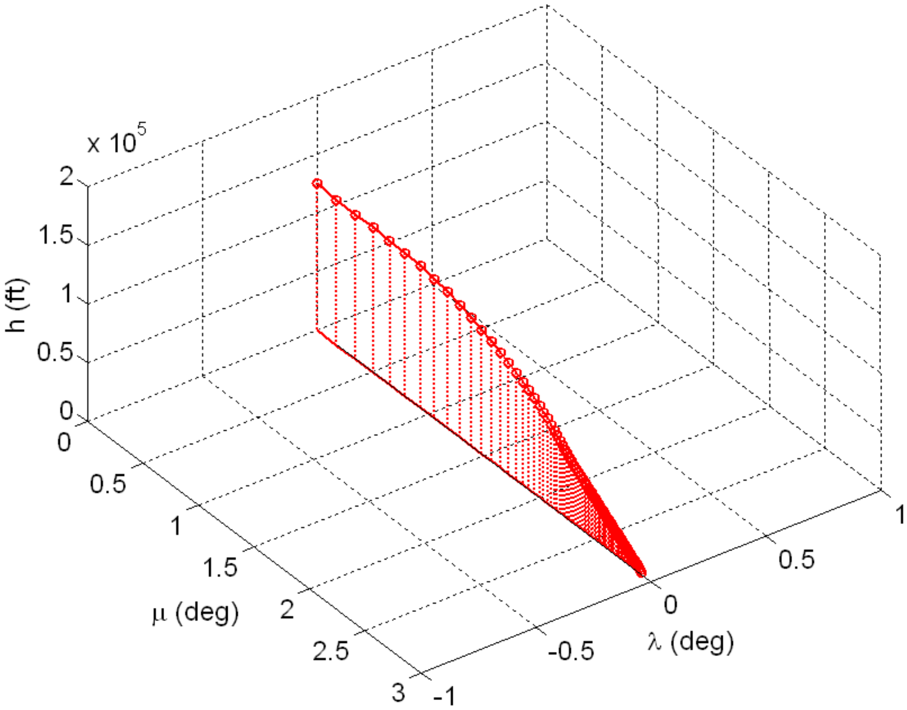


Figure 6.52 Max DR 3D Flight Trajectory Profile.

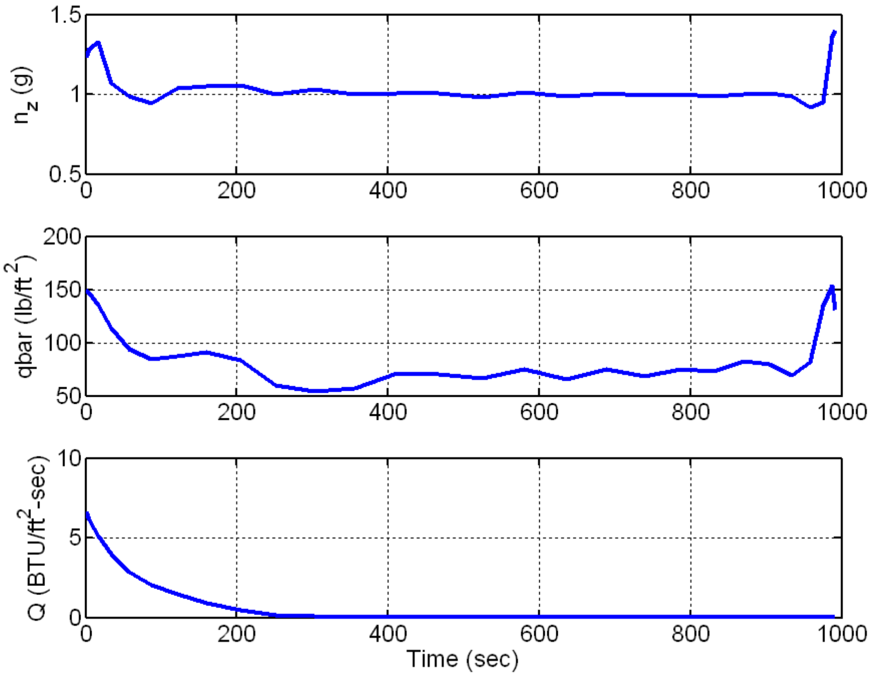


Figure 6.53 Max DR Path Constraint Functions.

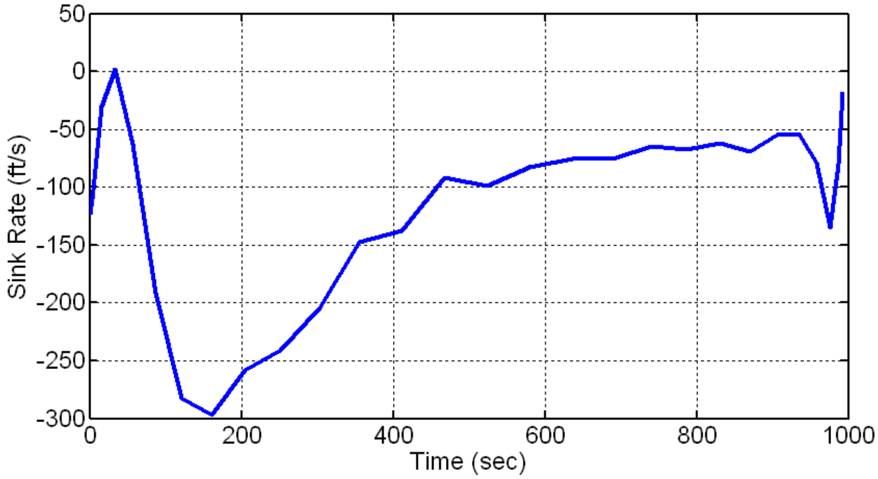


Figure 6.54 Max DR Sink Rate Profile.

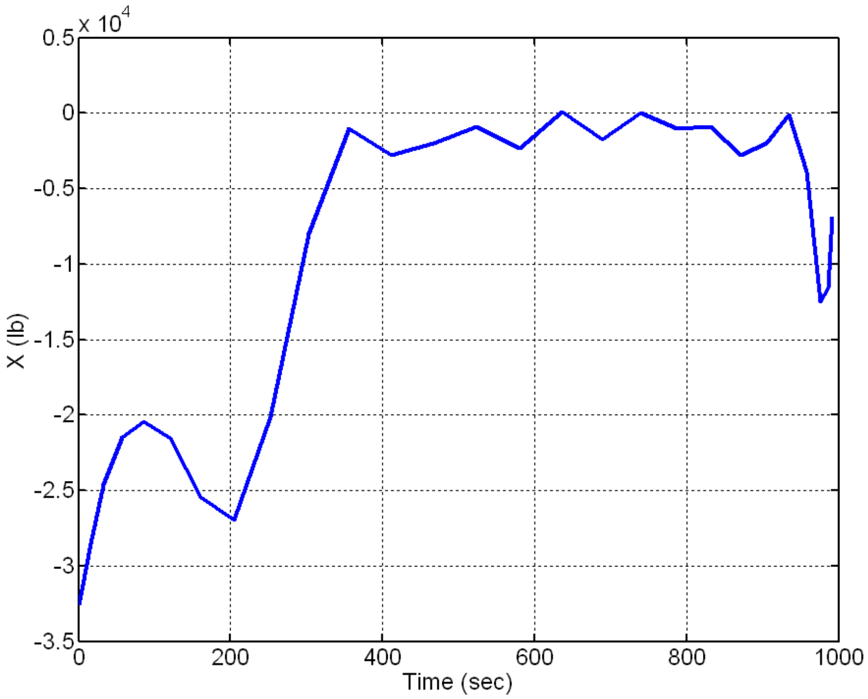


Figure 6.55   Max DR Axial Force History.

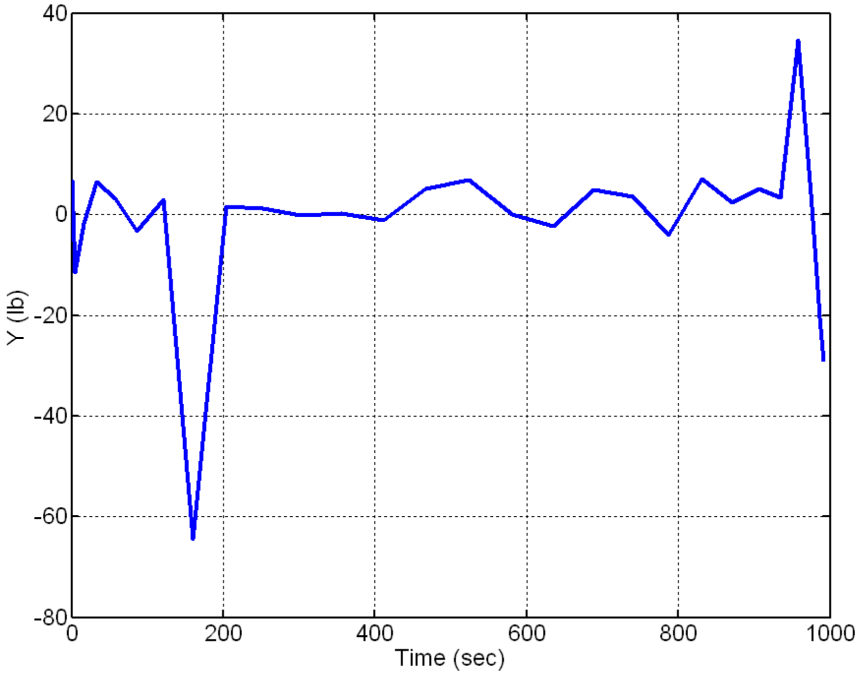


Figure 6.56   Max DR Side Force History.

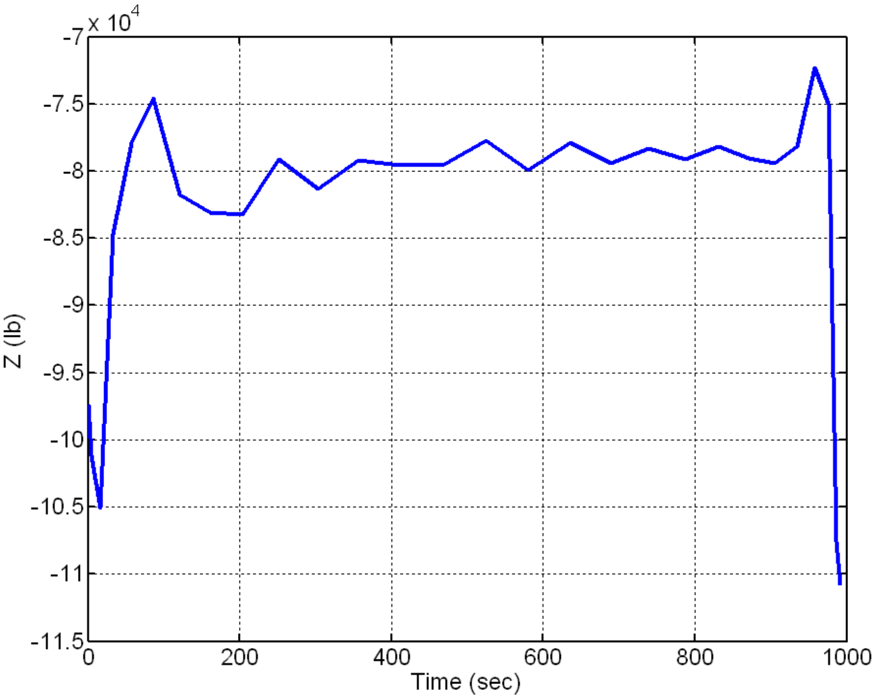


Figure 6.57   Max DR Normal Force History.

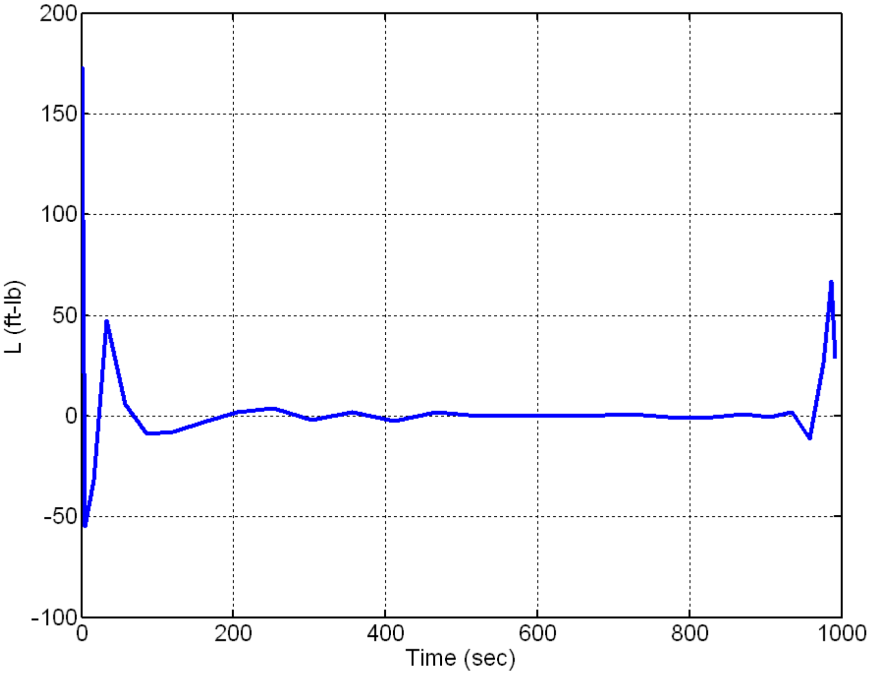


Figure 6.58   Max DR Rolling Moment History.

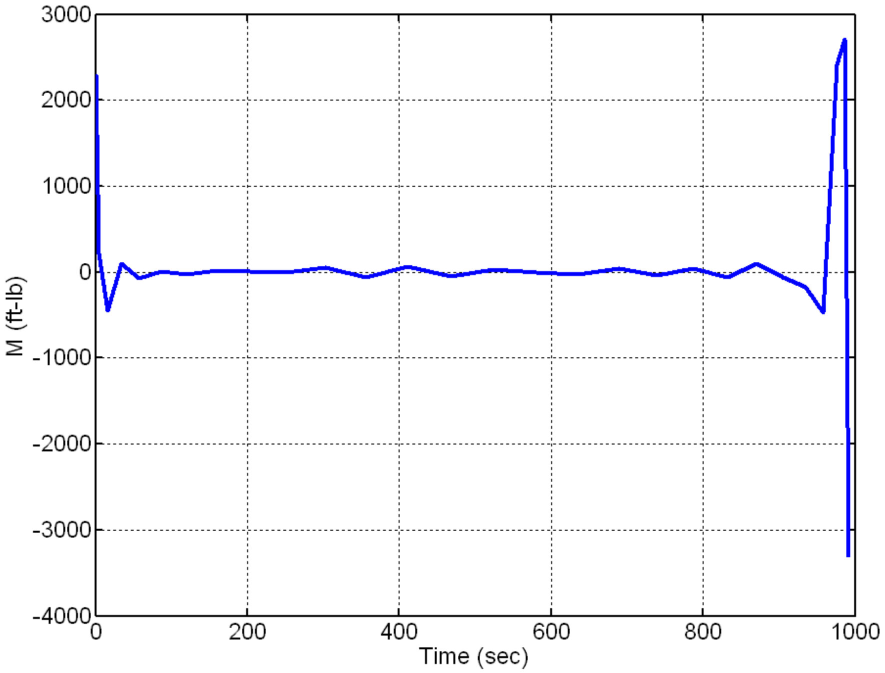


Figure 6.59 Max DR Pitching Moment History.

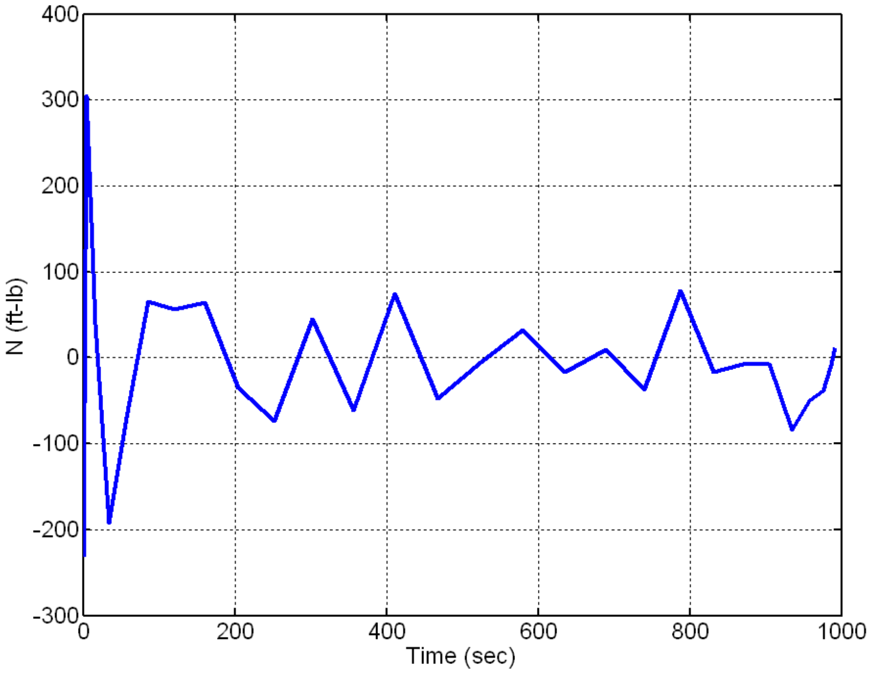


Figure 6.60 Max DR Yawing Moment History.

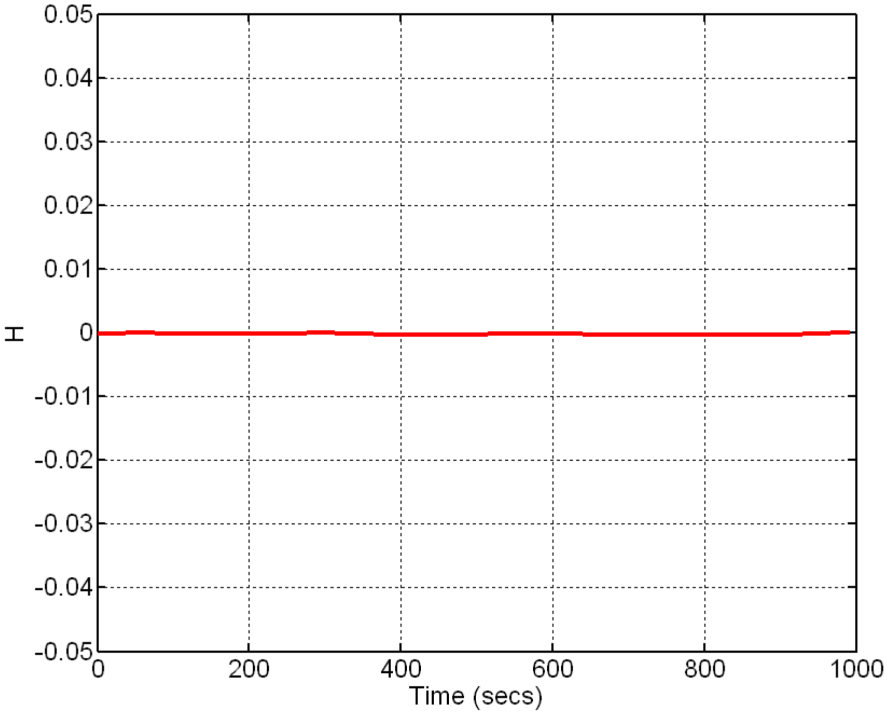


Figure 6.61 Max DR Hamiltonian Function.

As shown in Figure 6.45, Figure 6.51, and Figure 6.52, the max-range cost for this solution is approximately 2.908 deg in longitude (~174.48 nm) with a total flight time of approximately 992 sec. This solution not only computes approximately 74% faster than the standard model, but also provides a better cost. Figure 6.50 verifies that the bank angle and sideslip angle are approximately zero for the duration of the trajectory.

6. Problem MS: Modified Spherical-Earth Model

a. Problem Formulation

Similar to the standard spherical-earth model (Problem SS), a modified spherical-earth model (Problem MS) is employed. The only two differences in this model are the exploitation of the full, flight angle relationships as described in Chapter III and the relaxed requirement for coordinated turns, i.e., nonzero sideslip angle. With this, the equivalent, analytical flight angle relationships are given by Eqs. (3.16) and (3.18) and repeated here

$$\begin{aligned}\beta &= \operatorname{asin}\left[\cos\gamma(\sin\theta\sin\phi\cos(\xi\text{-}\psi)+\cos\phi\sin(\xi\text{-}\psi))\text{-}\sin\gamma\cos\theta\sin\phi\right]\\ \alpha &= \operatorname{acos}\left[(\cos\gamma(\cos\theta\cos(\xi\text{-}\psi))+\sin\gamma\sin\theta)/\cos\beta\right]\end{aligned} \tag{5.69}$$

which are again bounded by implementing the corresponding path constraints

$$\begin{aligned} -10 \le \alpha(\cdots,t) \le 50 \\ -1.5 \le \beta(\cdots,t) \le 1.5 \end{aligned} \tag{5.70}$$

This time, the constraint on β is not omitted. Similarly, an analytical expression for the bank angle is deduced from the flight angle DCM of Eq. (3.12),

$$\sigma\text{=acos}\left[(\cos\alpha\cos\theta\cos\phi\text{+}\sin\alpha\sin\theta)/\cos\gamma\right] \tag{5.71}$$

and constrained according to

$$-80\,\text{deg} \le \sigma(\cdots,t) \le 80\,\text{deg} \tag{5.72}$$

b. Results and Analysis

The following results were generated by bootstrapping from 20 to 20 nodes which takes approximately 335 sec.

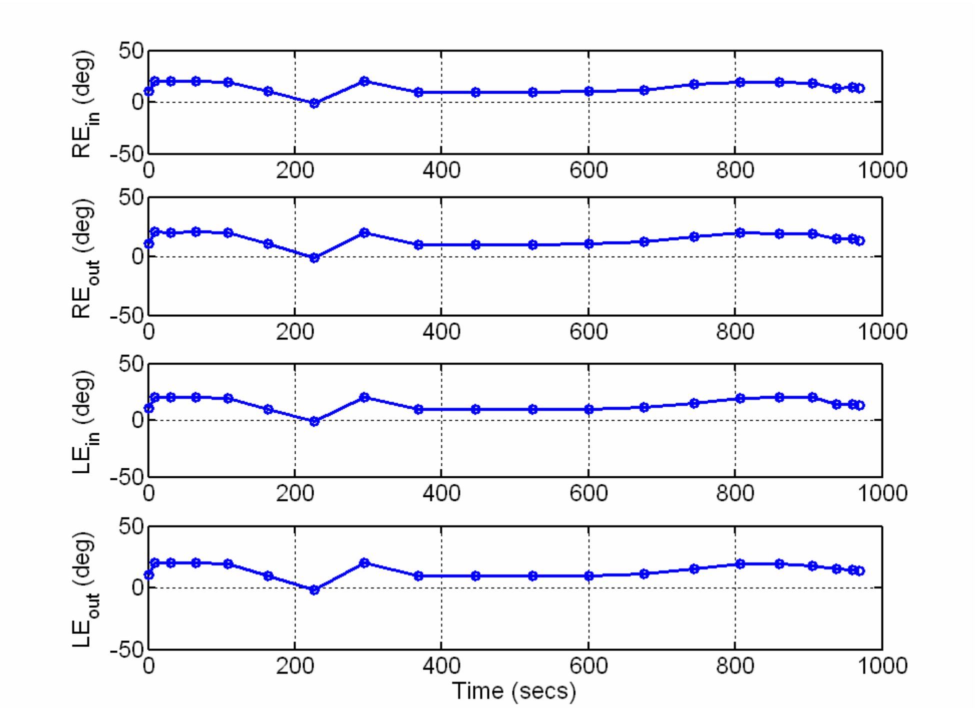


Figure 6.62 Max DR Elevon Controls.

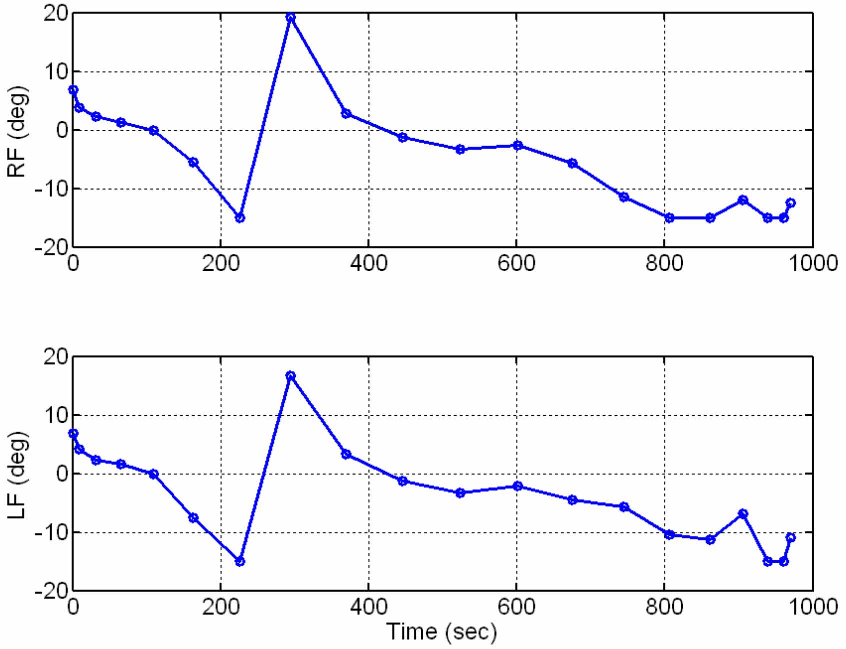


Figure 6.63   Max DR Flap Controls.

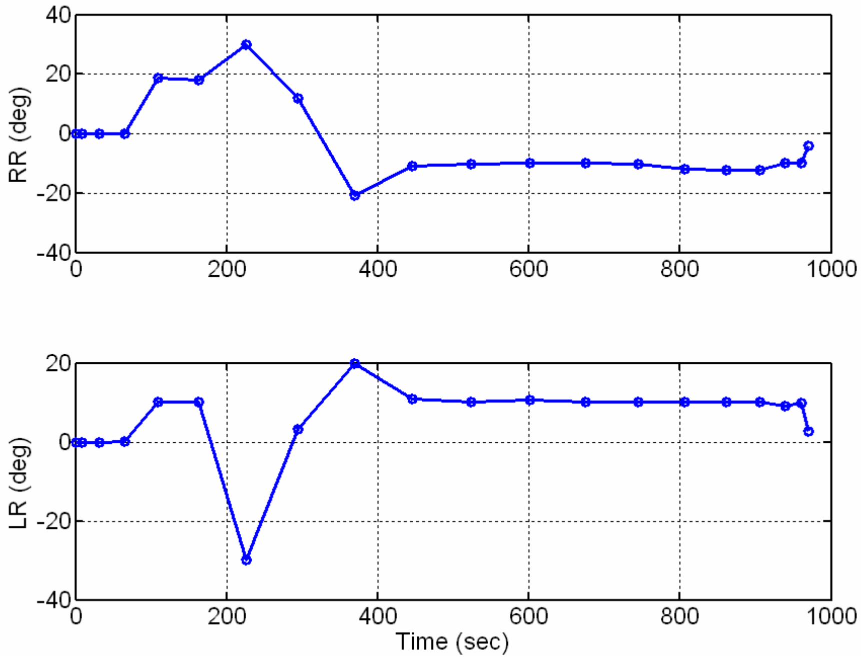


Figure 6.64   Max DR Rudder Controls.

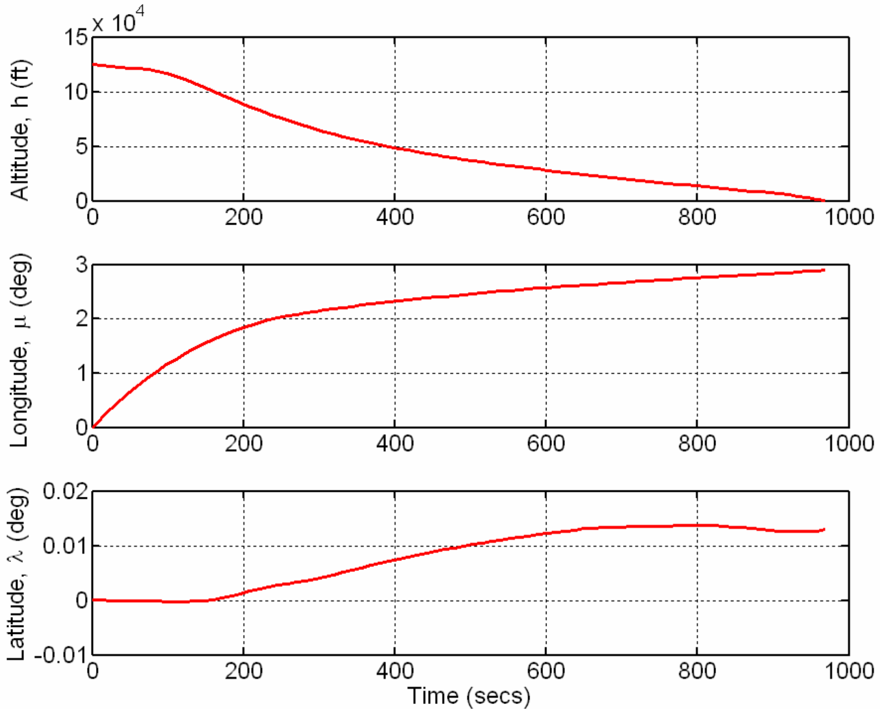


Figure 6.65 Max DR Position Response.

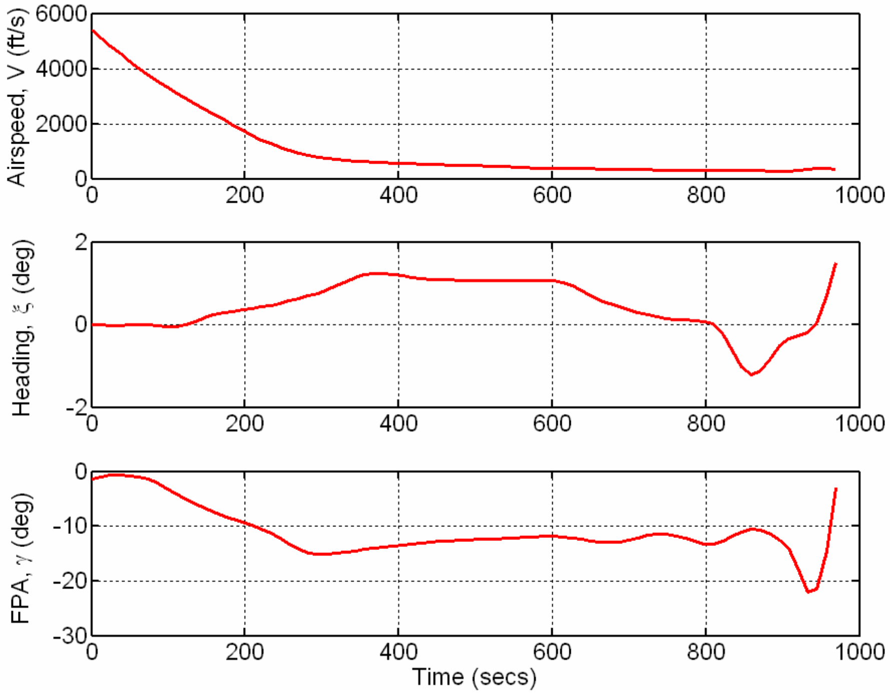


Figure 6.66 Max DR Velocity Response.

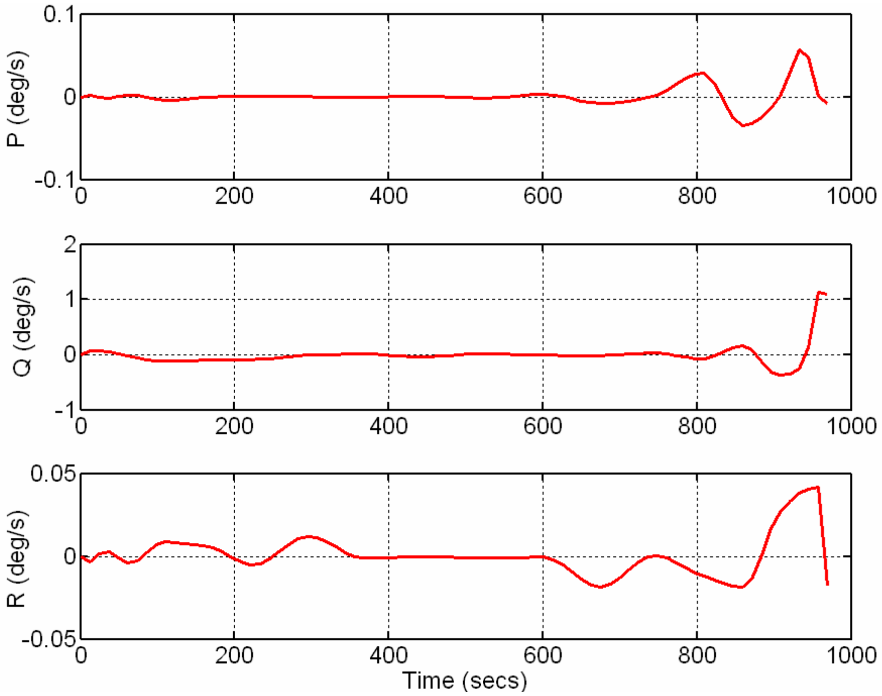


Figure 6.67 Max DR Body Rate Response (degrees).

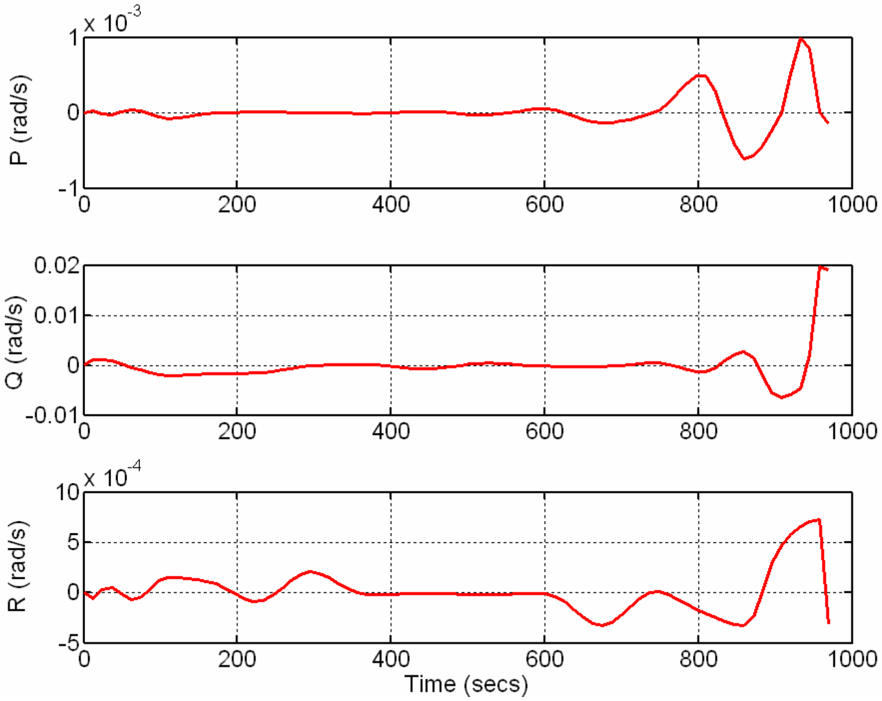


Figure 6.68 Max DR Body Rate Response (Radians).

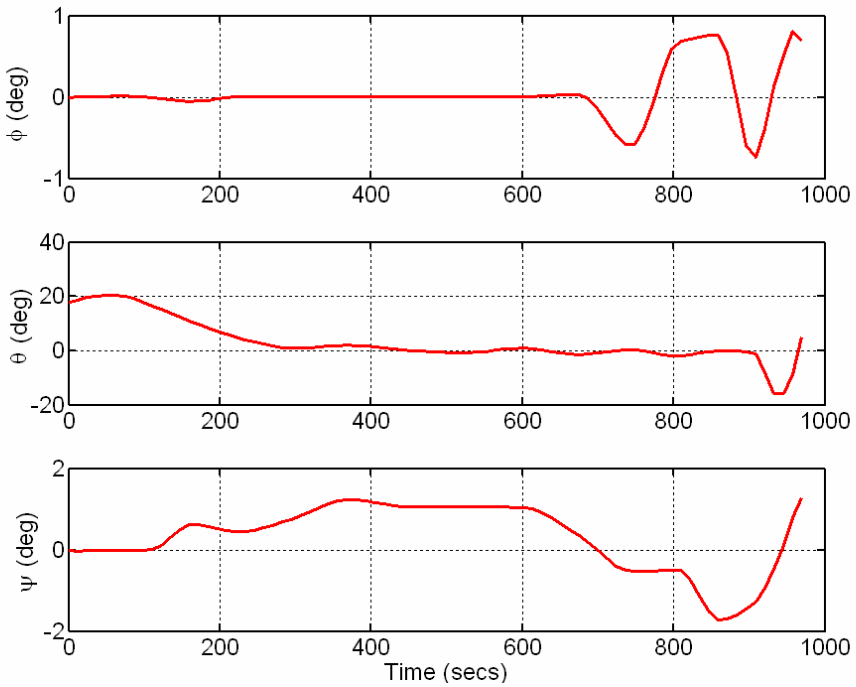


Figure 6.69 Max DR Euler Angle Response.

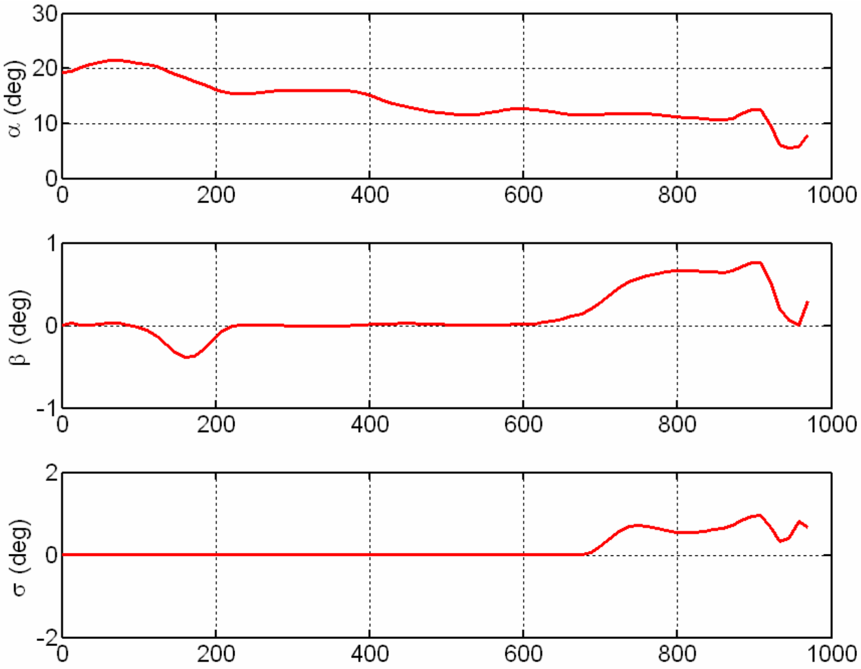


Figure 6.70 Max DR Aero Angle Response.

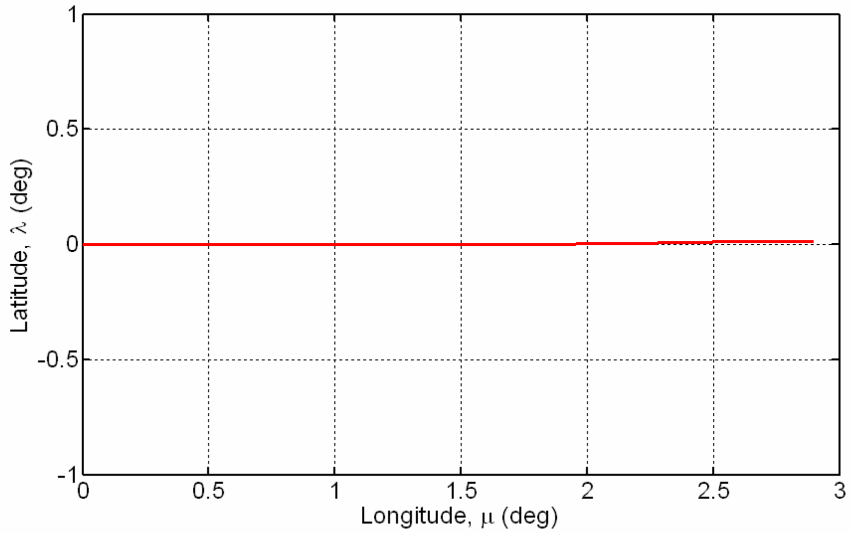


Figure 6.71 Max DR 2D Ground Track.

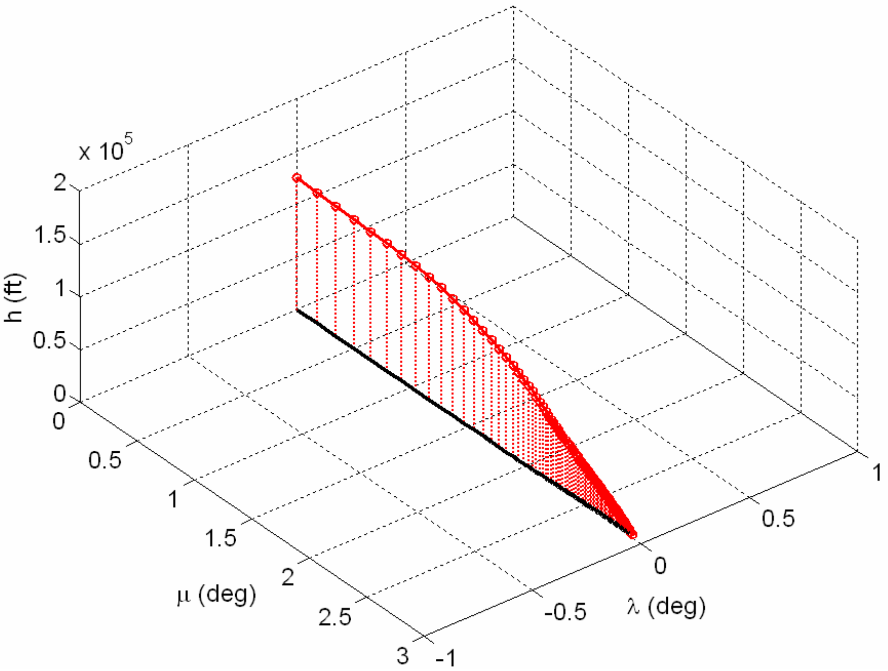


Figure 6.72 Max DR 3D Flight Trajectory Profile.

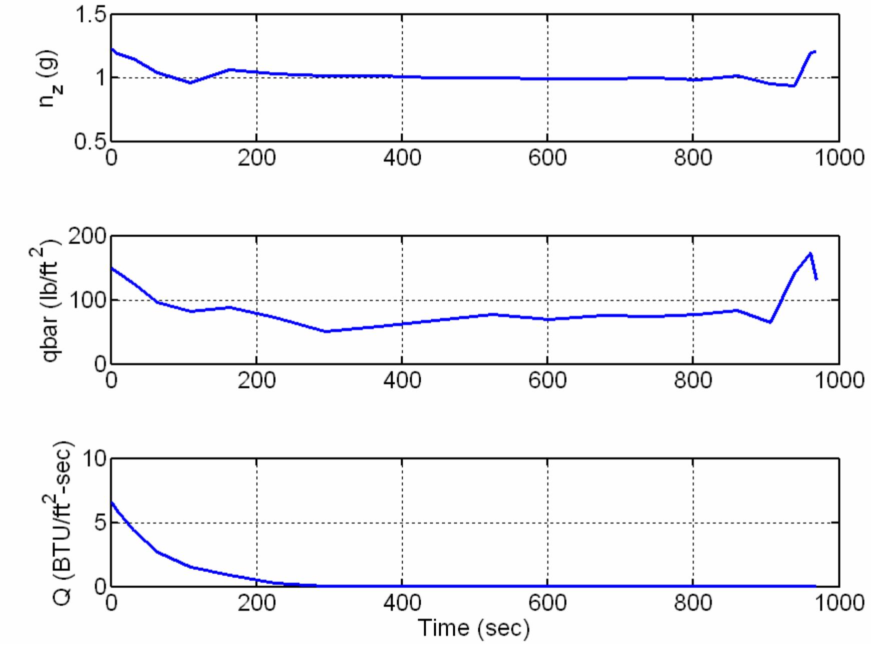


Figure 6.73 Max DR Path Constraint Functions.

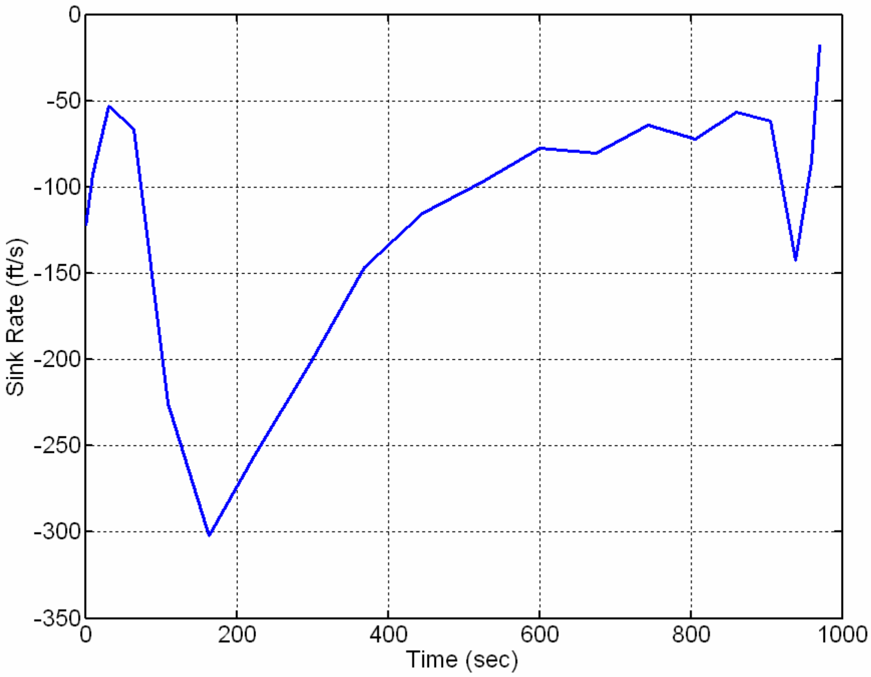


Figure 6.74 Max DR Sink Rate Profile.

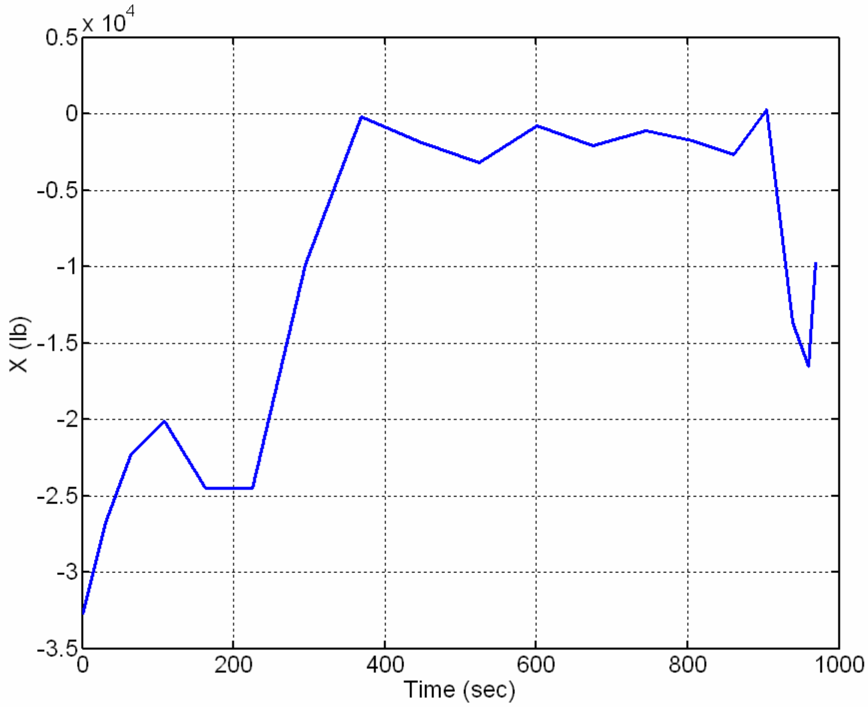


Figure 6.75   Max DR Axial Force History.

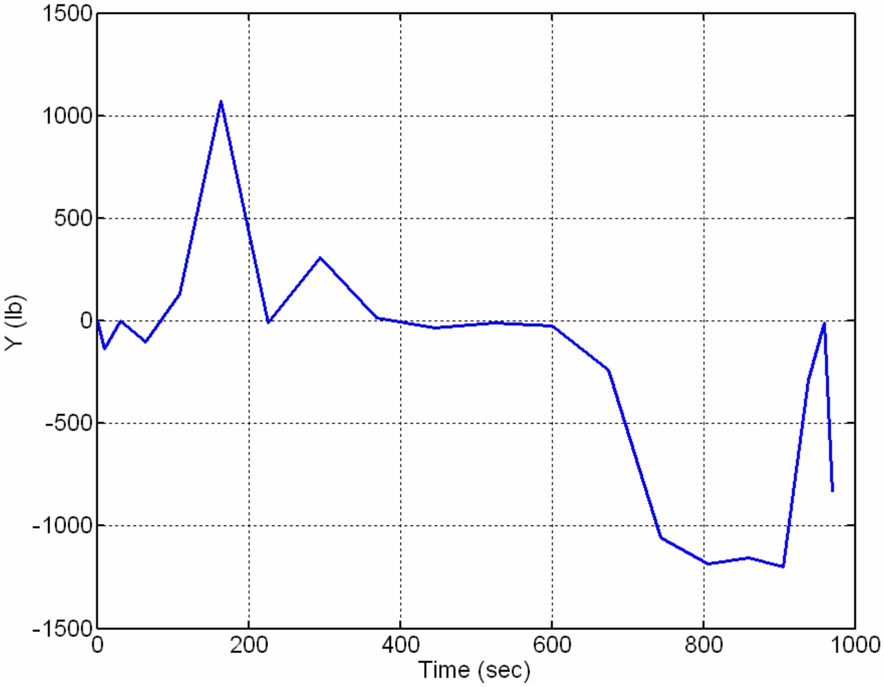


Figure 6.76   Max DR Side Force History.

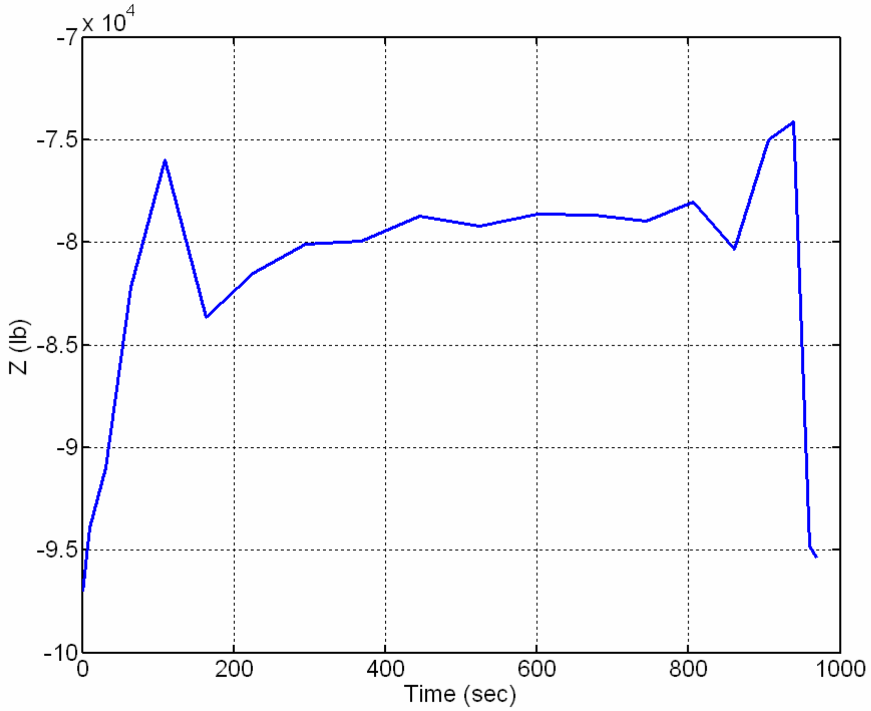


Figure 6.77 Max DR Normal Force History.

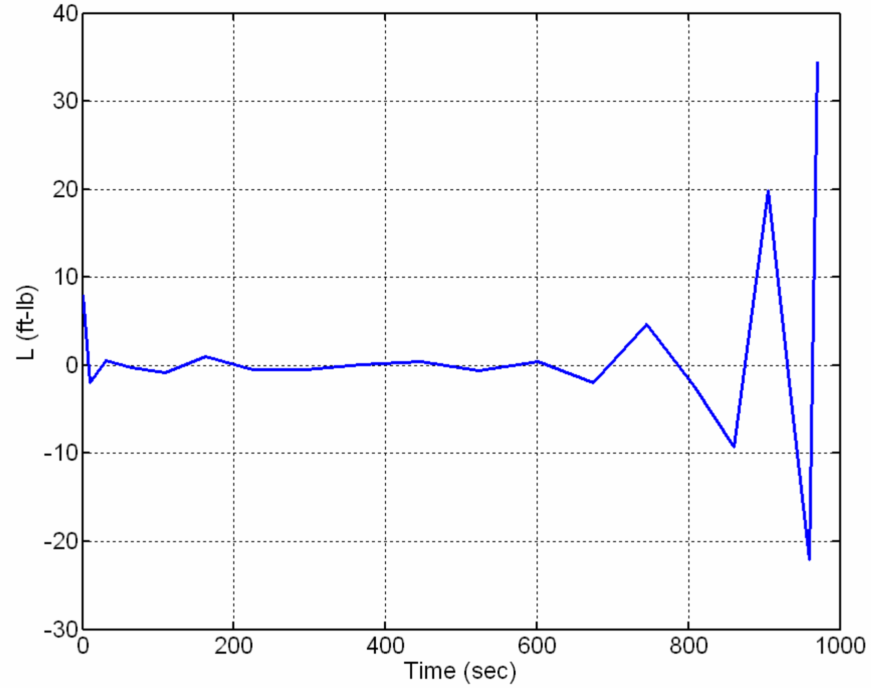


Figure 6.78 Max DR Rolling Moment History.

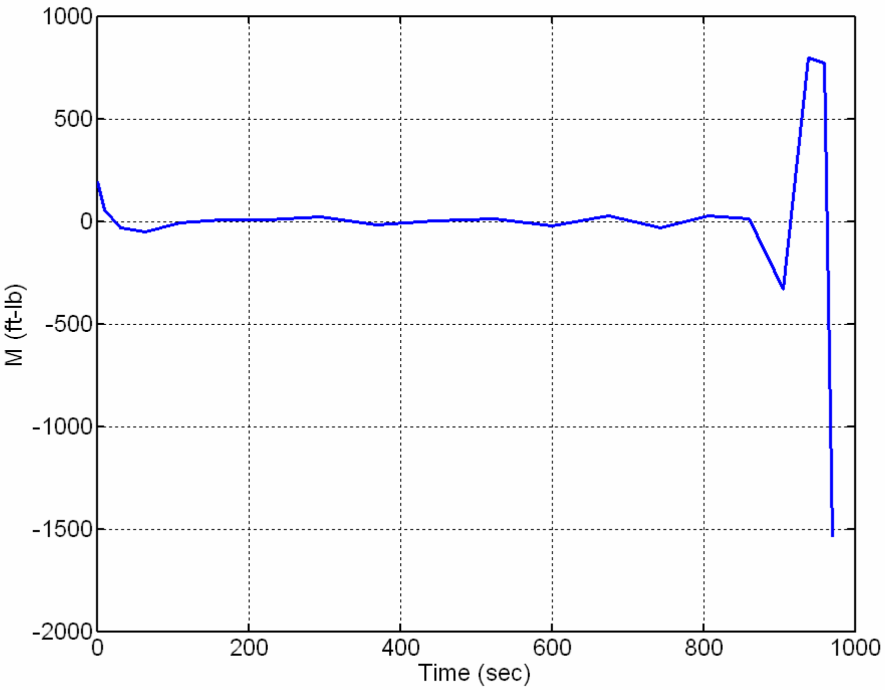


Figure 6.79   Max DR Pitching Moment History.

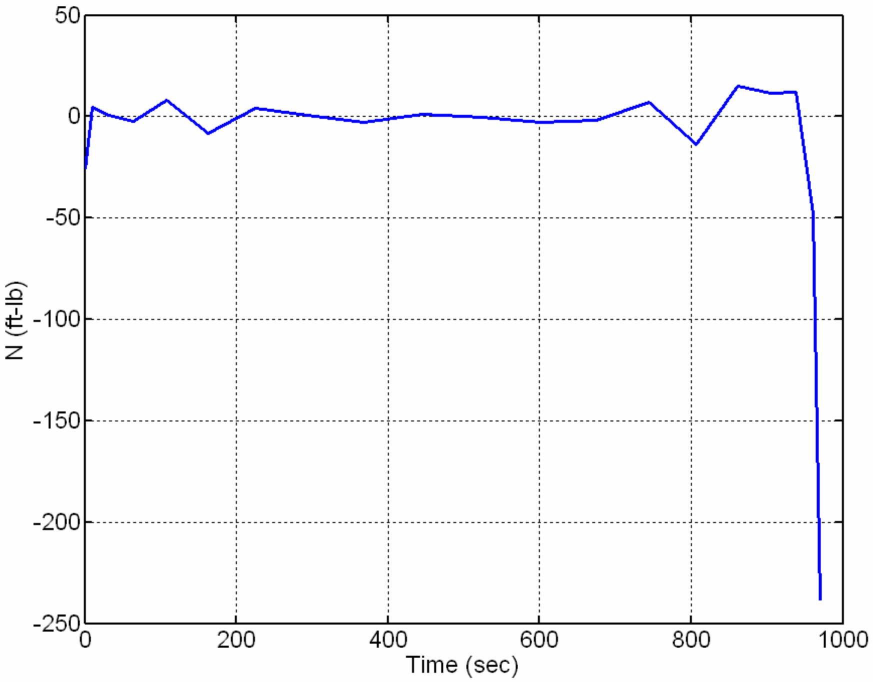


Figure 6.80   Max DR Yawing Moment History.

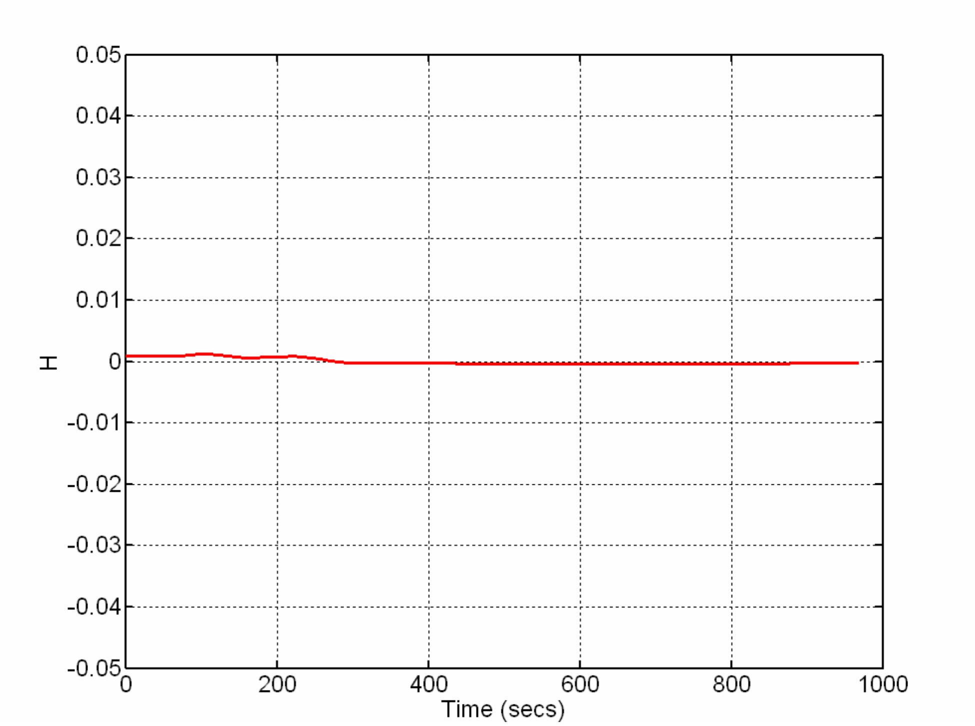


Figure 6.81 Max DR Hamiltonian Function.

As expected, these results are very similar to those for Problem SS since the rolling/banking angles are relatively small throughout the trajectory. Interestingly, Figure 6.70 indicates nonzero sideslip and bank angles at the end of the trajectory. Although this will typically reduce the downrange distance due to an increase in drag, this maneuver may be necessary to satisfy the endpoint conditions. The maximum downrange for this solution is 2.899 deg in longitude (~173.95 nm), as shown in Figure 6.65, Figure 6.71, and Figure 6.72.

The following plots show the results for a Max crossrange (CR) case.

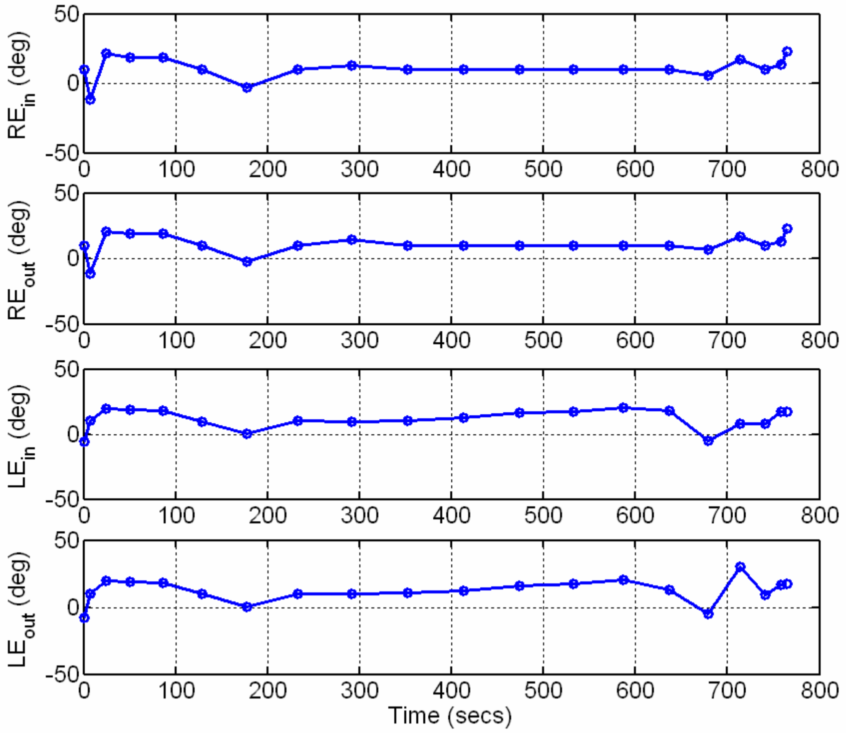


Figure 6.82 Max CR Elevon Controls.

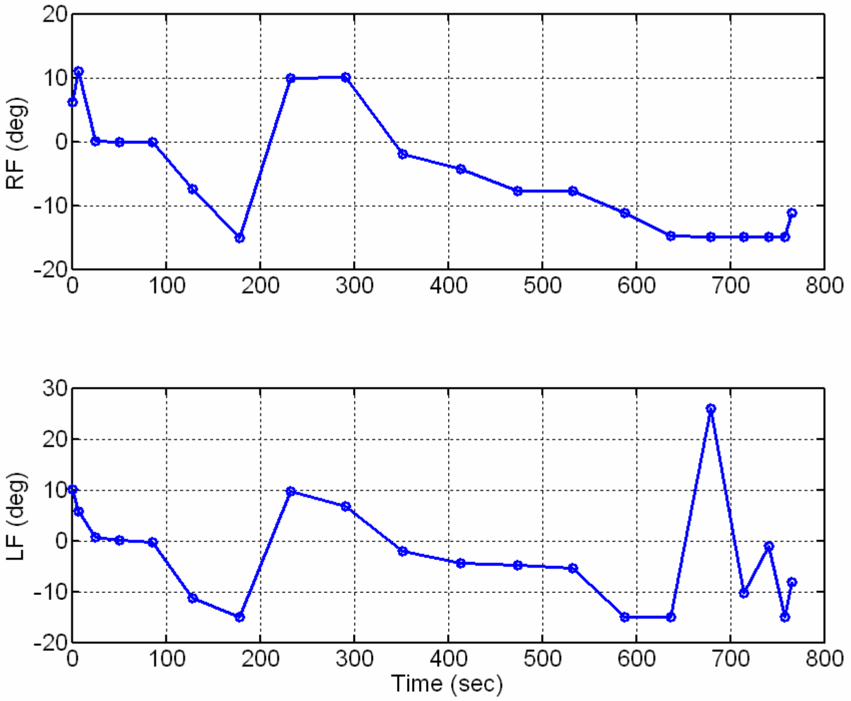


Figure 6.83 Max CR Flap Controls.

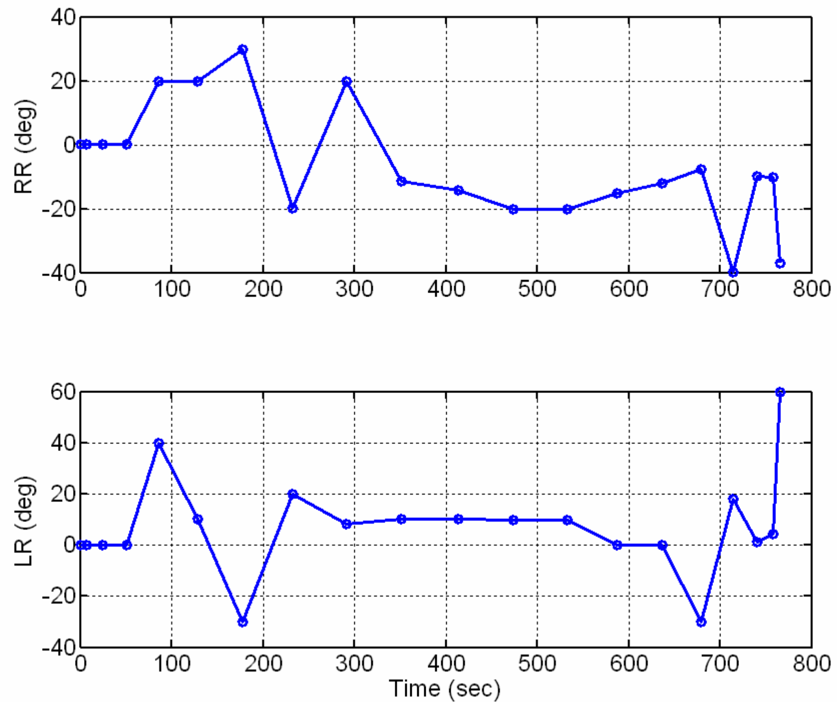


Figure 6.84   Max CR Rudder Controls.

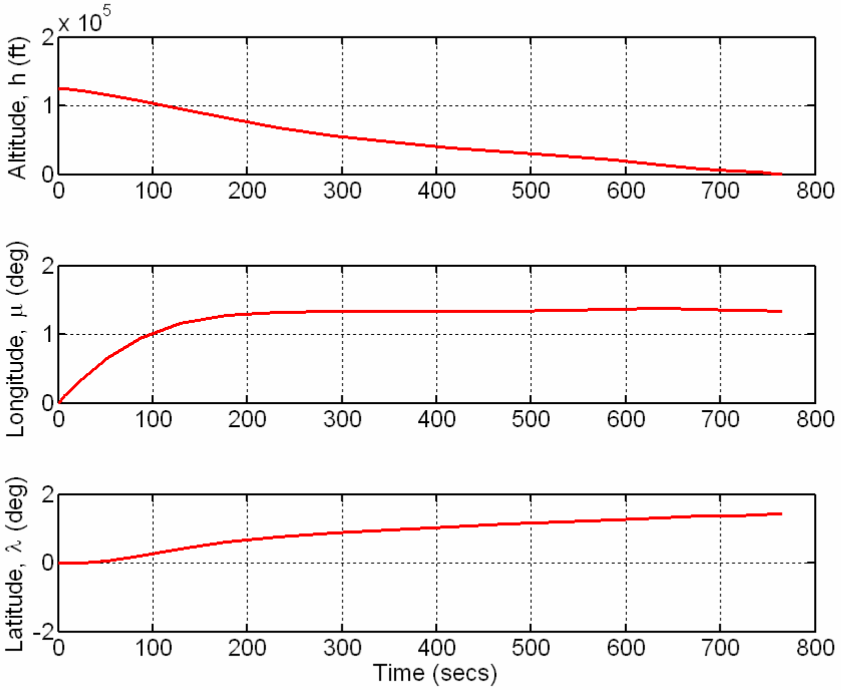


Figure 6.85   Max CR Position Response.

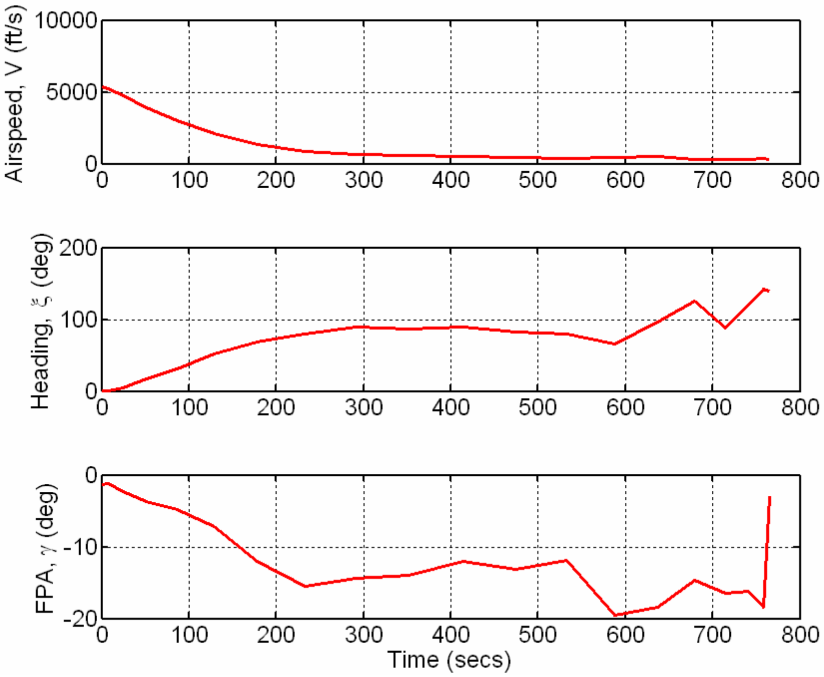


Figure 6.86 Max CR Velocity Response.

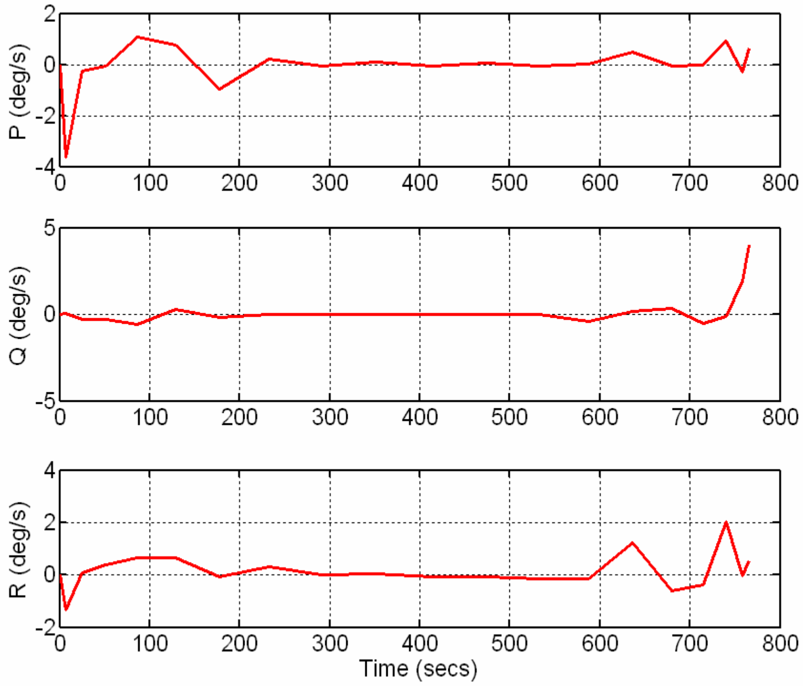


Figure 6.87 Max CR Body Rate Response (degrees).

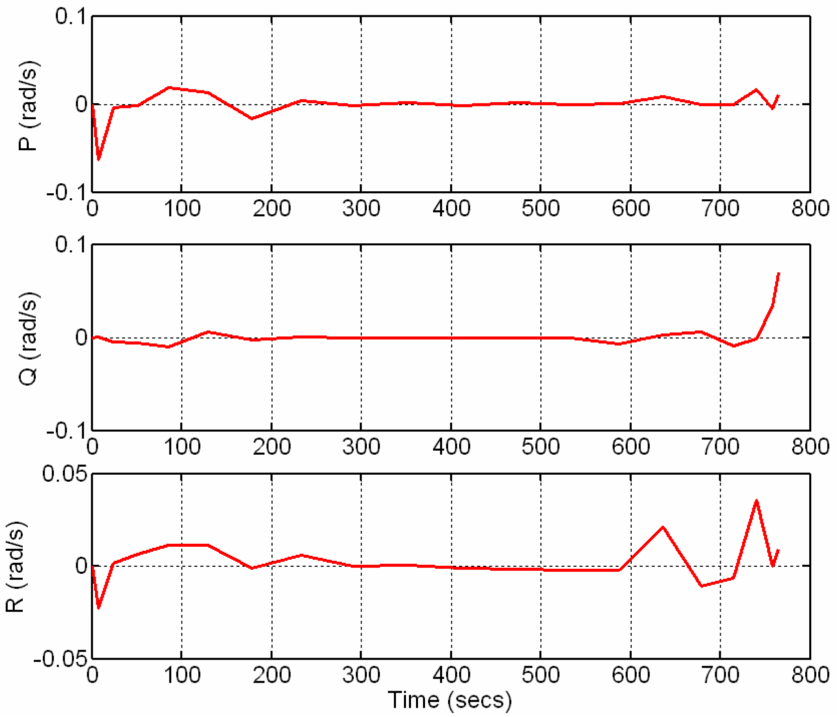


Figure 6.88 Max CR Body Rate Response (Radians).

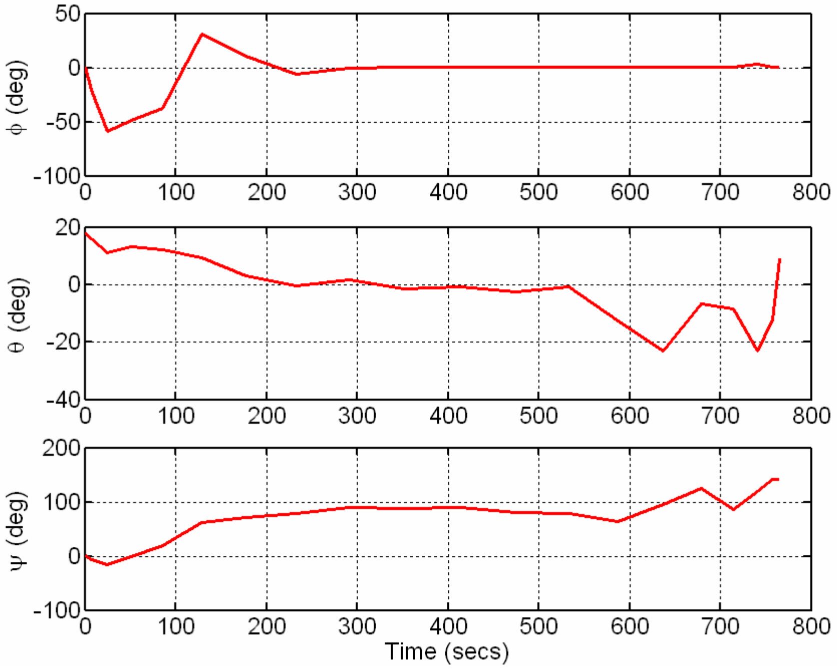


Figure 6.89 Max CR Euler Angle Response.

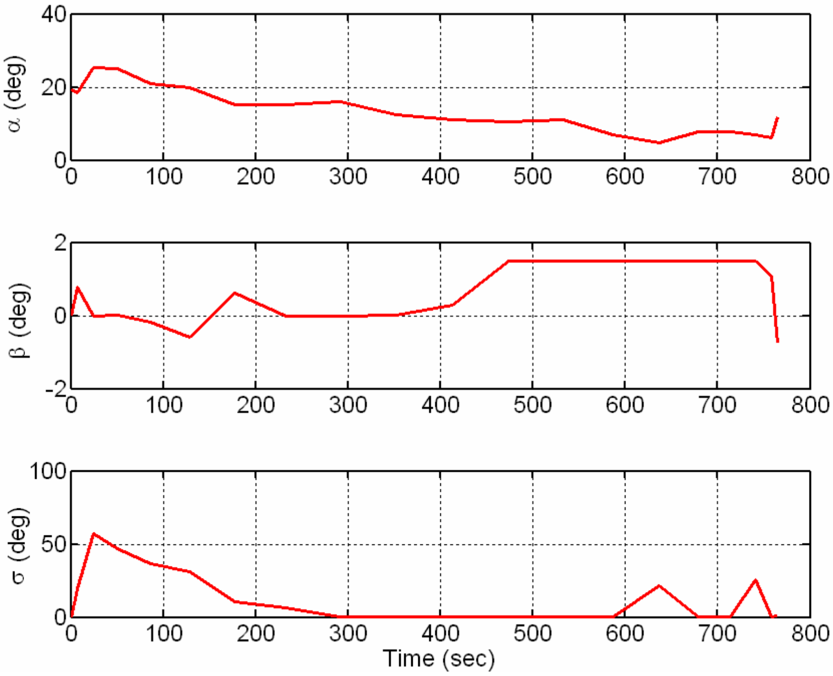


Figure 6.90 Max CR Aero Angle Response.

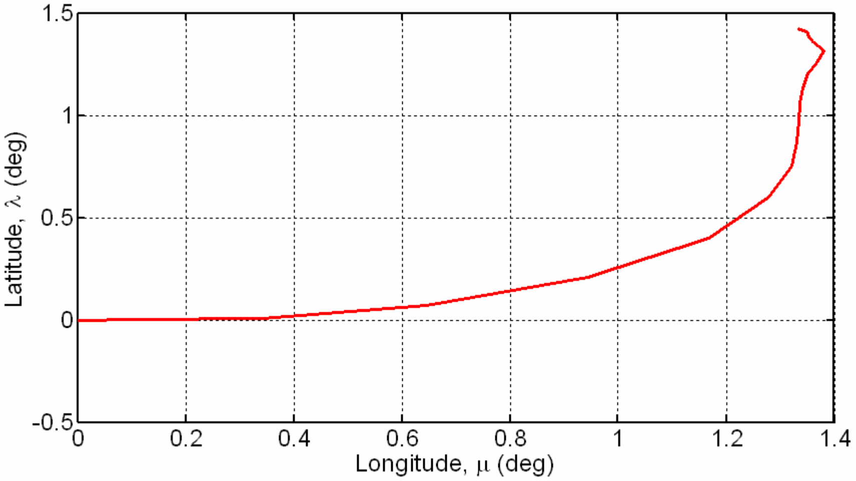


Figure 6.91 Max CR 2D Ground Track.

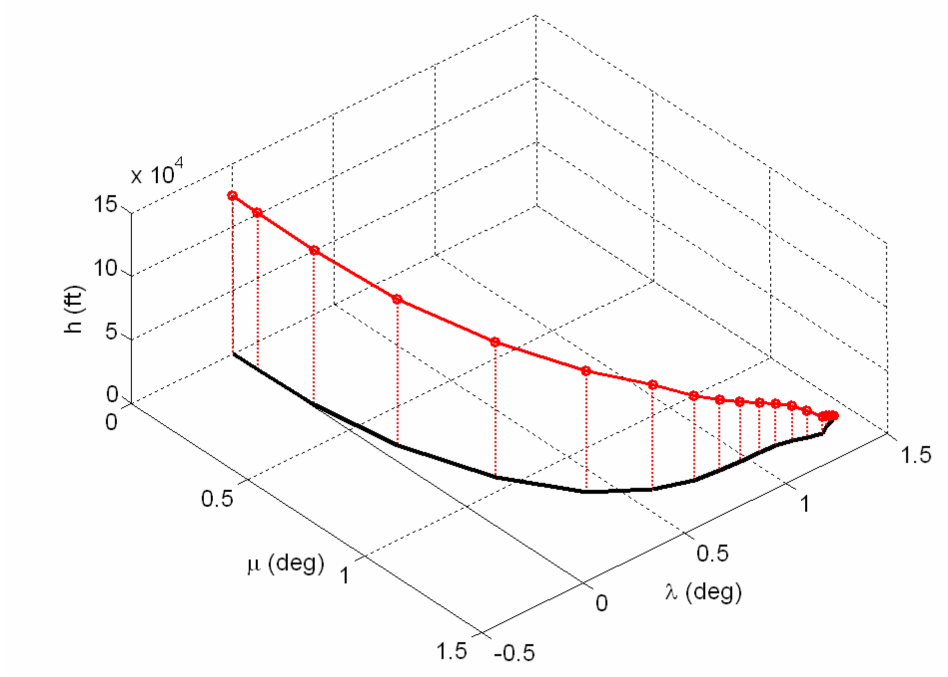


Figure 6.92 Max CR 3D Flight Trajectory Profile.

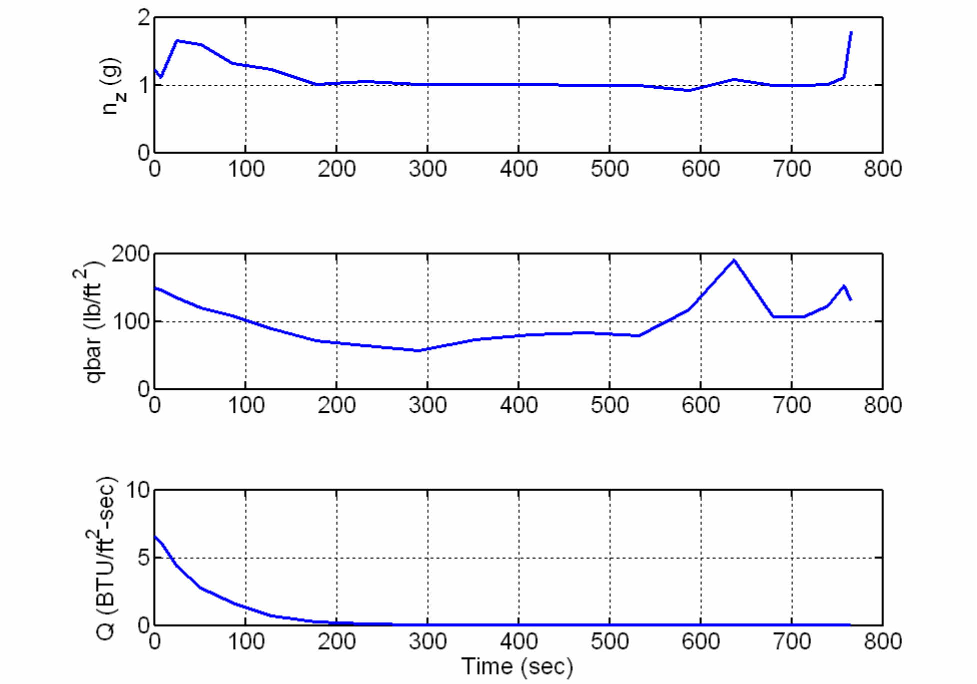


Figure 6.93 Max CR Path Constraint Functions.

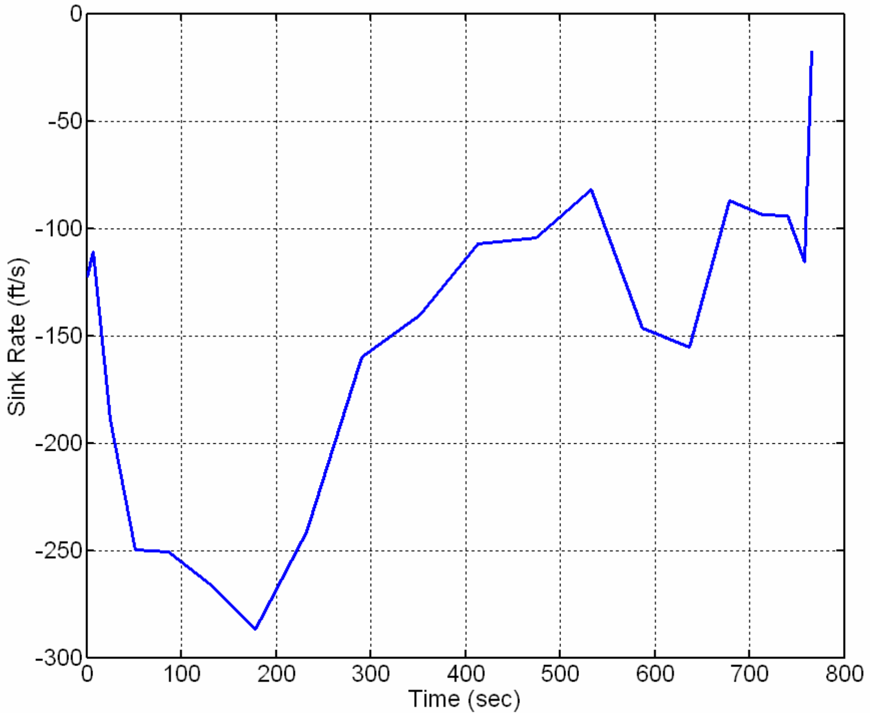


Figure 6.94 Max CR Sink Rate Profile.

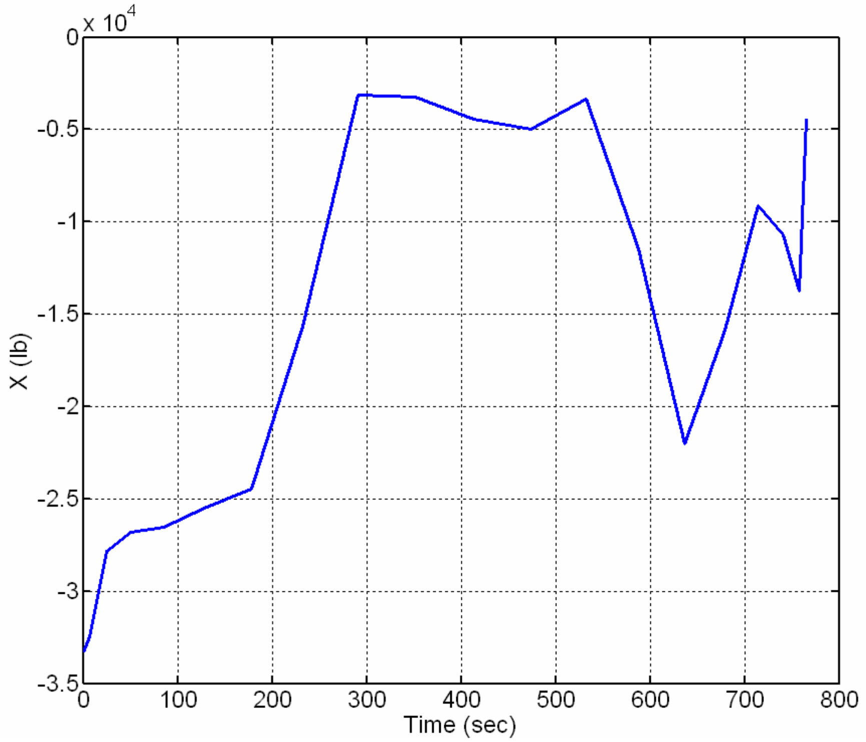


Figure 6.95 Max CR Axial Force History.

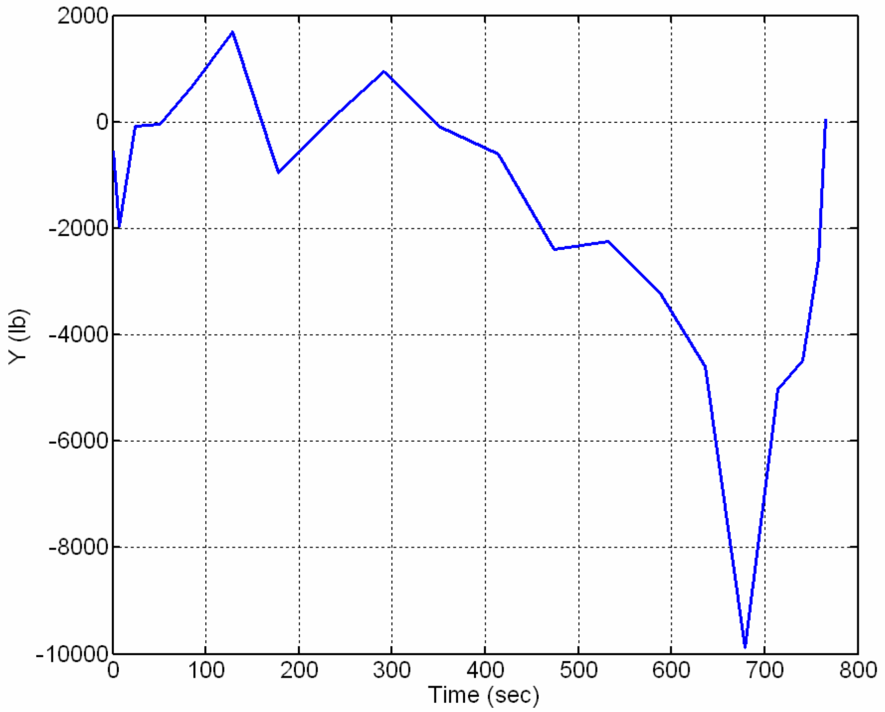


Figure 6.96 Max CR Side Force History.

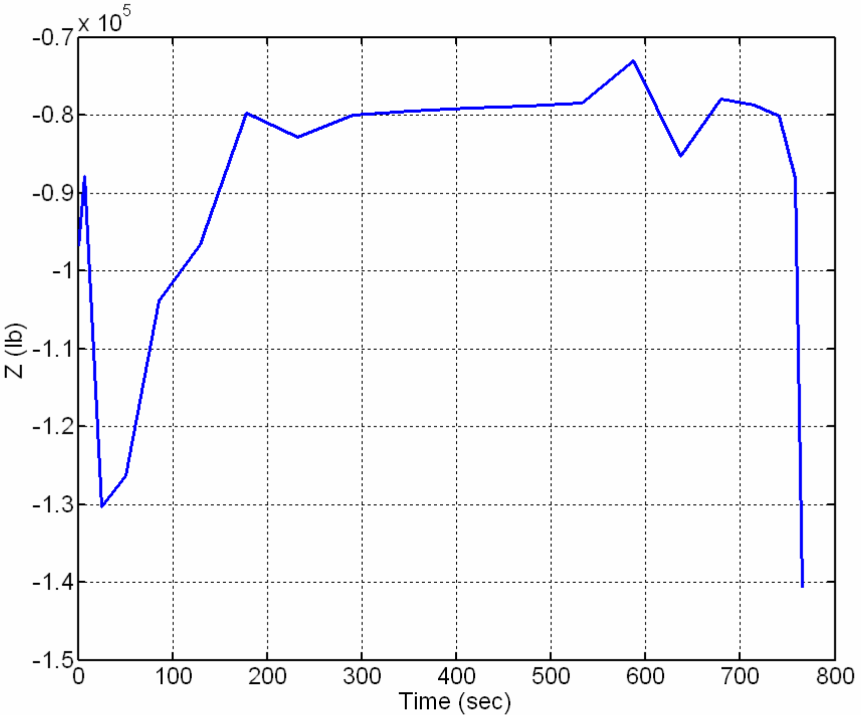


Figure 6.97 Max CR Normal Force History.

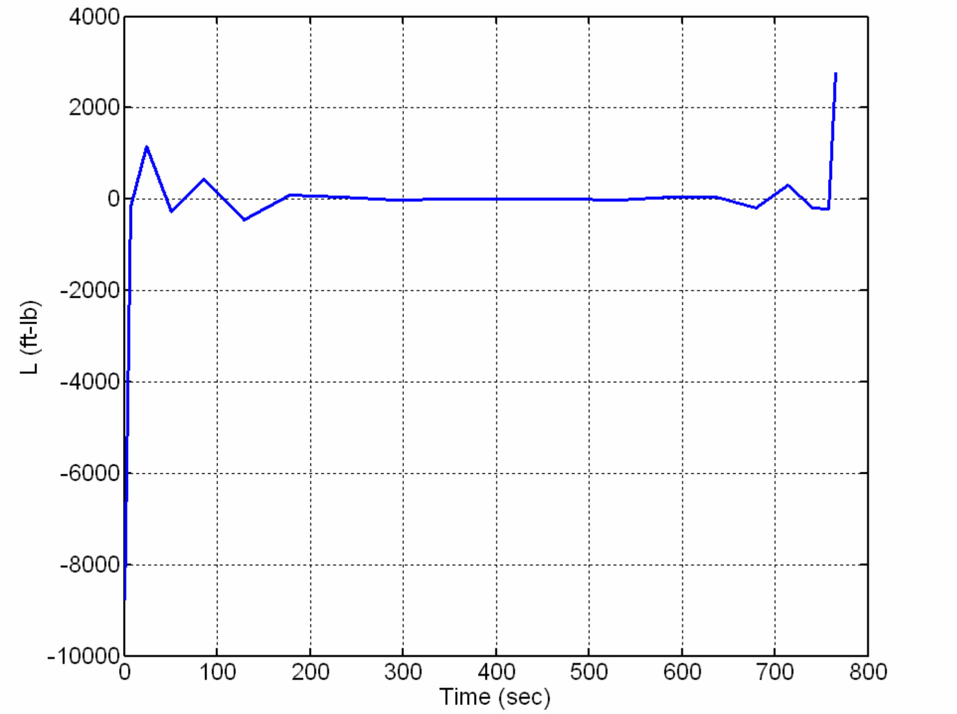


Figure 6.98   Max CR Rolling Moment History.

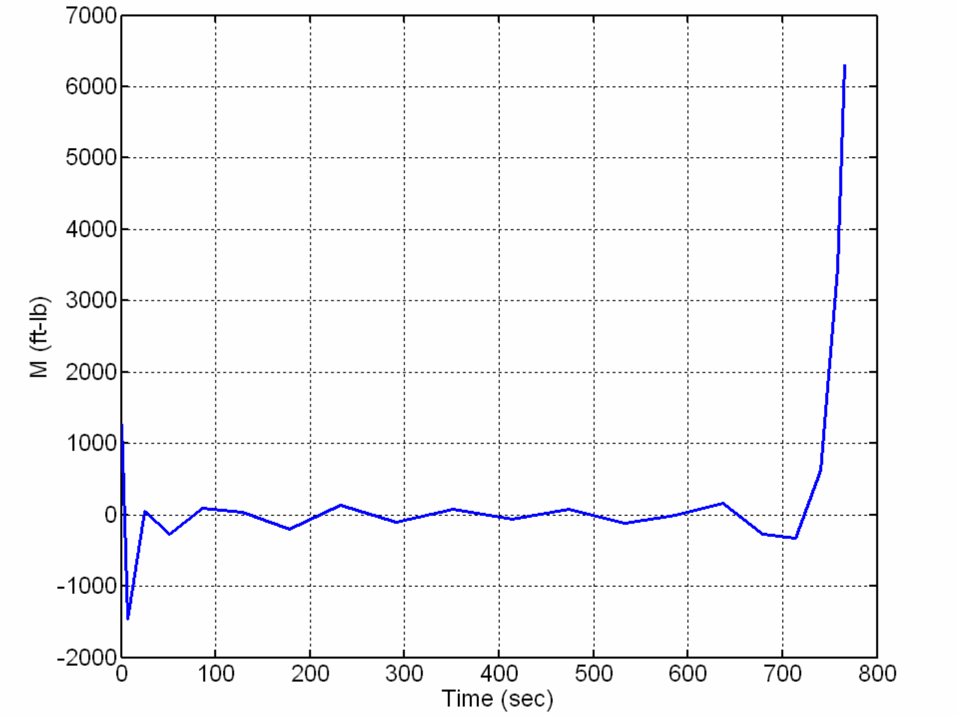


Figure 6.99   Max CR Pitching Moment History.

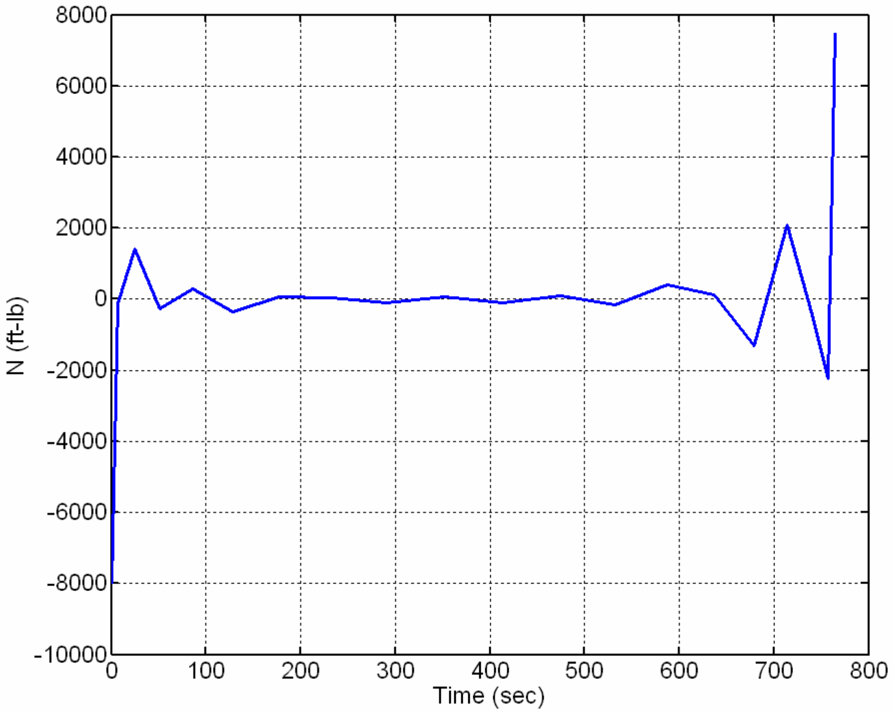


Figure 6.100 Max CR Yawing Moment History.

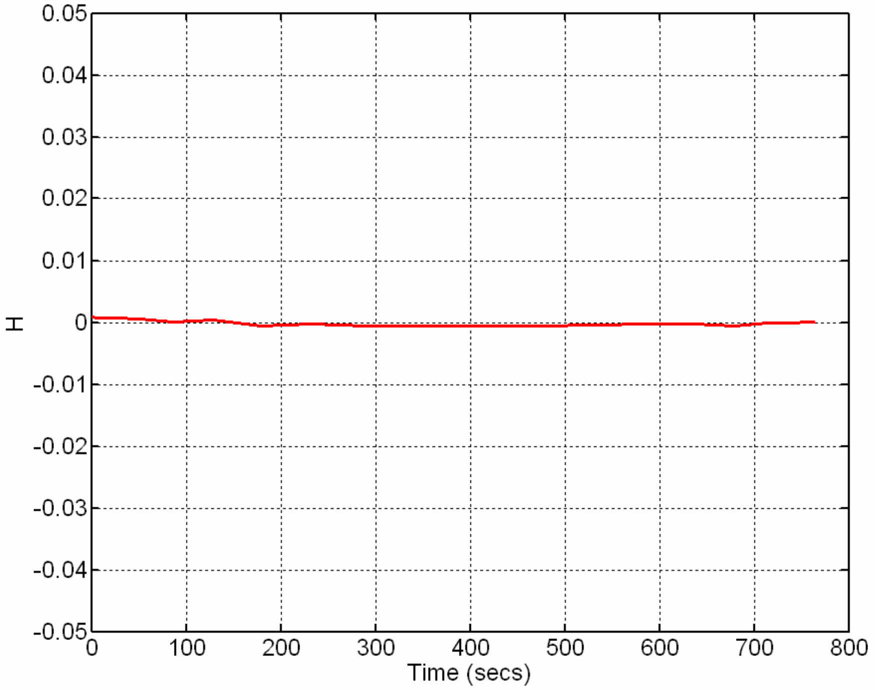


Figure 6.101 Max CR Hamiltonian Function.

The maximum crossrange for this solution is 1.425 deg in latitude (~ 85.5 nm) as shown by Figure 6.85, Figure 6.91, and Figure 6.92. As expected, Figure 6.89 and Figure 6.90 show a large roll and bank angle maneuver for the first 100 sec or so, until the heading is approximately 90 deg when they "level out." Surprisingly, the sideslip angle also increases near the end of the trajectory. A shown in Figure 6.90, the sideslip angle constraint is active from about 475 sec to 750 sec. Again, this may be a result of the vehicle requiring energy management in order to satisfy the endpoint conditions.

7. 6-DOF versus 3-DOF Solutions

The following results compare the maximum downrange case for the 3-DOF and the 6-DOF trajectory optimization solutions. Since the 3-DOF model used in Chapter IV assumes zero sideslip, it is appropriate to compare Problem SS which also forces zero sideslip.

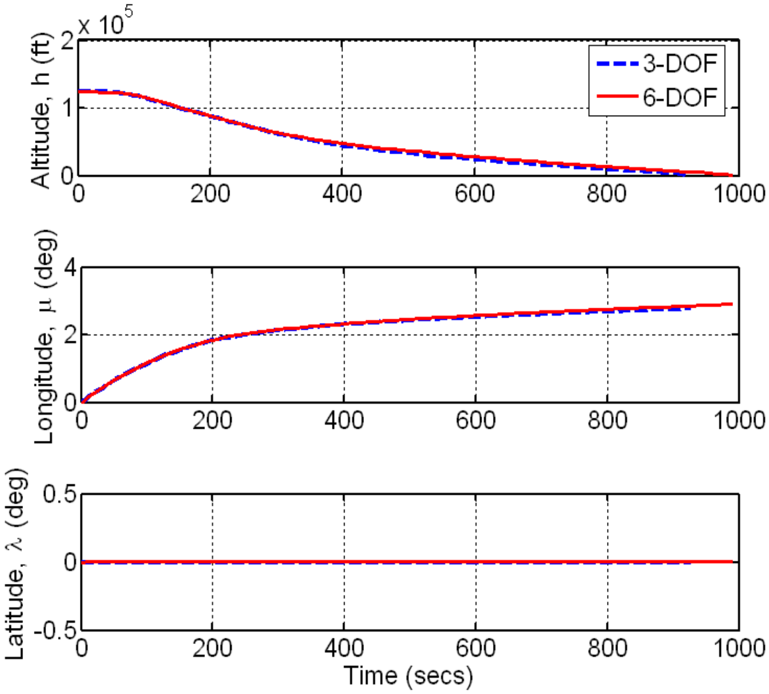


Figure 6.102 3-DOF vs. 6-DOF Response (Position) for Max DR.

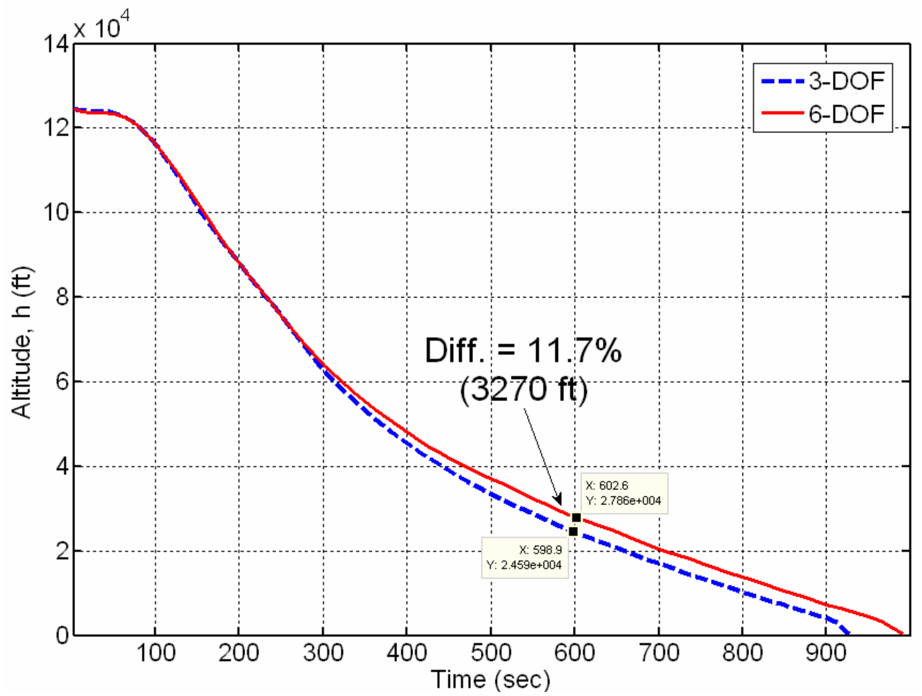


Figure 6.103 A Closer Look at 3-DOF vs. 6-DOF Altitude Profile.

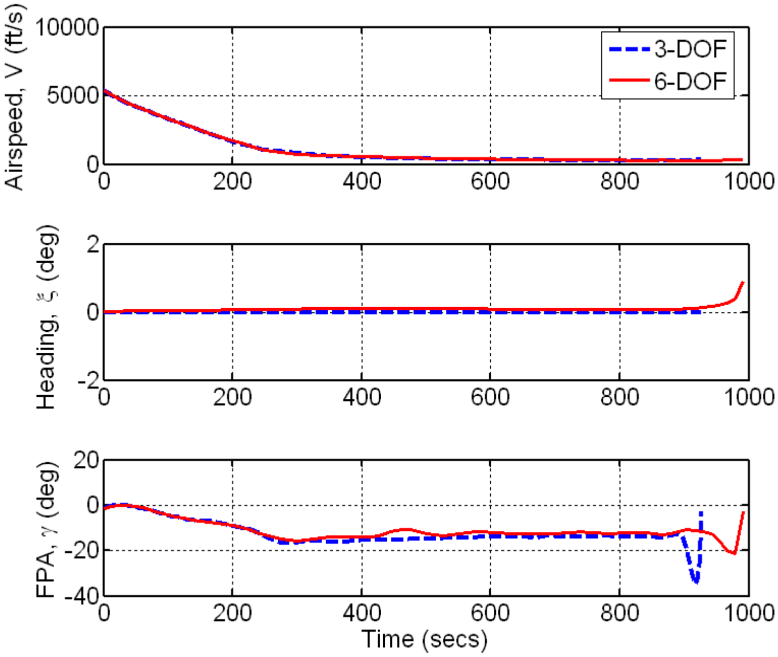


Figure 6.104 3-DOF vs. 6-DOF Response (Velocity) for Max DR.

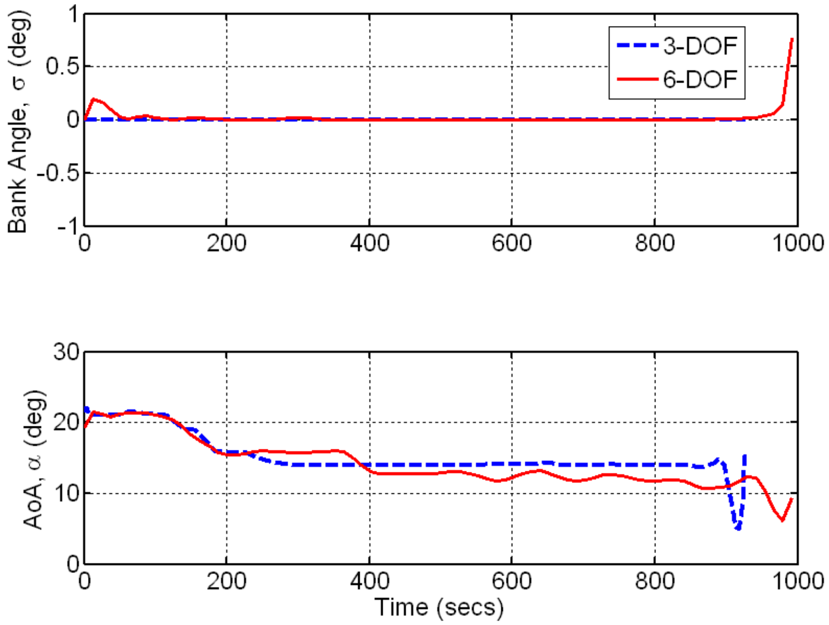


Figure 6.105 3-DOF vs. 6-DOF AoA and Bank Angle for Max DR.

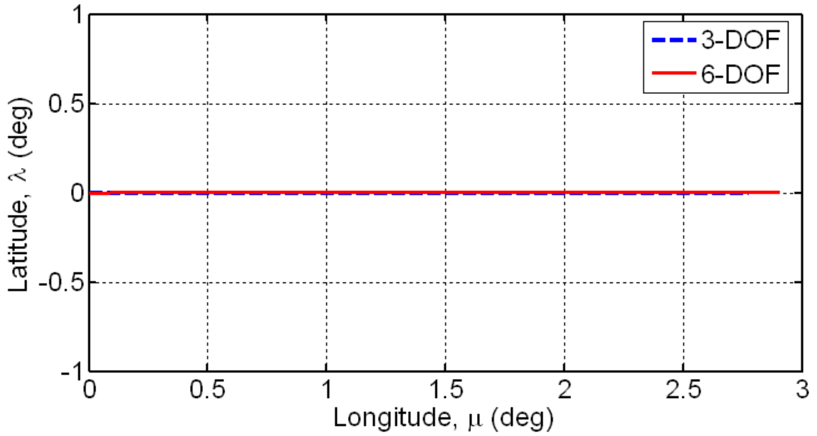


Figure 6.106 3-DOF vs. 6-DOF 2D Ground Track for Max DR.

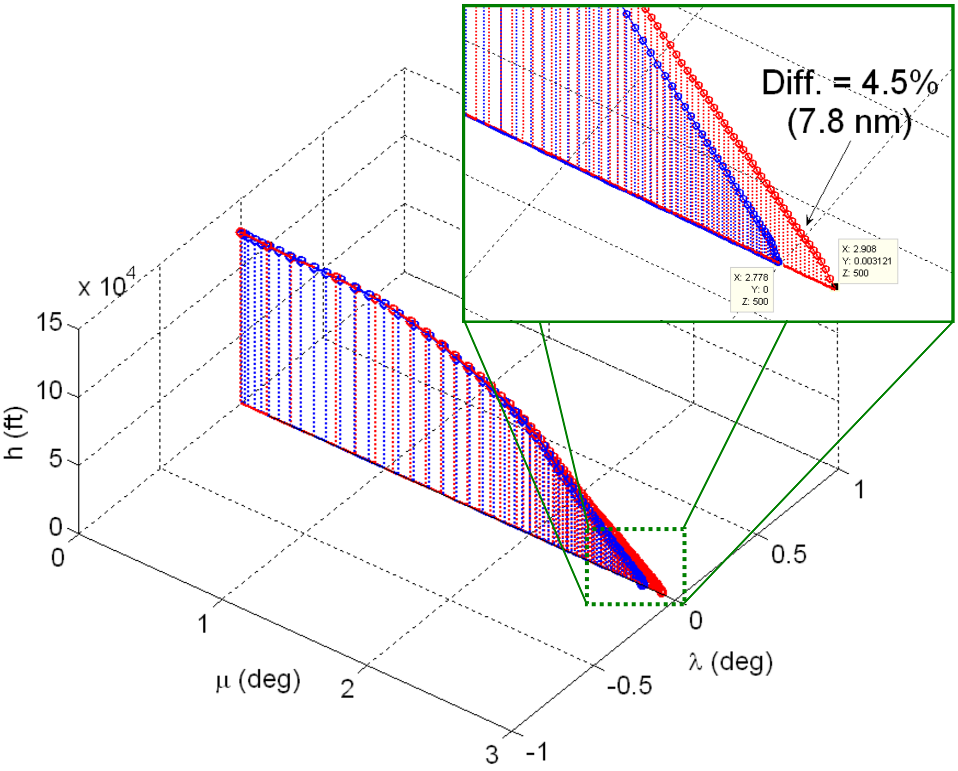


Figure 6.107 3-DOF vs. 6-DOF 3D Flight Trajectory for Max DR.

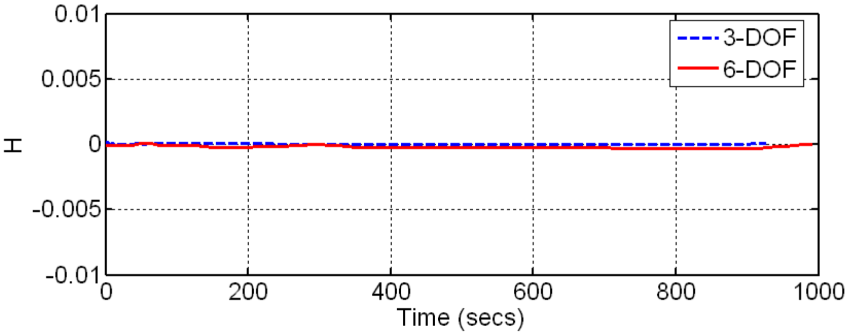


Figure 6.108 3-DOF vs. 6-DOF Max DR Hamiltonian Function.

Figure 6.102 appears to show very little difference between the 3-DOF and 6-DOF max DR cases; however, a closer look in Figure 6.103 and Figure 6.107 reveals a difference of 11.7 % (3270 ft) in altitude at 600 sec and 4.5 % (7.8 nm) in downrange distance. Additional comparisons to the 3-DOF trajectory are shown for the other 6-DOF models in Table 6.1 to Table 6.4.

Table 6.1 6-DOF vs. 3-DOF for Approx. Angles, Nonzero Beta (Unconstrained)

Case 1		Lon_f (nm)	Lat_f (nm)	Max Beta (deg)
6-DOF	Max DR	173.760	0.170	6.00
	Max CR	82.740	89.520	8.00
3-DOF	Max DR	166.680	0.000	0.00
	Max CR	88.860	71.880	0.00

Table 6.2 6-DOF vs. 3-DOF for Approx. Angles, Zero Beta

Case 2		Lon_f (nm)	Lat_f (nm)	Max Beta (deg)
6-DOF	Max DR	174.480	0.187	0.00
	Max CR	92.832	75.093	0.00
3-DOF	Max DR	166.680	0.000	0.00
	Max CR	88.860	71.880	0.00

Table 6.3 6-DOF vs. 3-DOF for Exact Angles, Nonzero Beta (Constrained)

Case 3		Lon_f (nm)	Lat_f (nm)	Max Beta (deg)
6-DOF	Max DR	173.940	0.776	0.77
	Max CR	79.980	85.500	1.50
3-DOF	Max DR	166.680	0.000	0.00
	Max CR	88.860	71.880	0.00

Table 6.4 6-DOF vs. 3-DOF Range Errors

	Case 1	Case 2	Case 3
Max DR Error (%)	4.075	4.470	4.174
Max CR Error (%)	19.705	4.279	15.930

Overall, the three different downrange cases result in approximately the same performance with an average error of 4.24 %. Note that the two cases that allowed nonzero sideslip resulted in a slightly smaller range. The crossrange performance had a higher error with an average of 13.3 %. The lowest error, indicated in Table 6.4, was for Case 2 that forced sideslip to be zero.

8. Feasibility Analysis

Confirming the feasibility of the 6-DOF results presented some challenges not encountered with the 3-DOF models. Unfortunately, the most suitable method for verifying the feasibility of the optimal control solution failed to work for the 6-DOF models that used the aerodynamic data (i.e., Problems SF, SS and MS). Note that for a successful feasibility test, the propagated results should converge to the DIDO solution within some acceptable error tolerance. As shown in the 3-DOF results, the only measurable errors were towards the end of the flight trajectory and were dependent on the accuracy of the solution (i.e., number of nodes). For all of the 6-DOF results (except the homotopy models that did not use the aerodynamic data), using the optimal control solution to propagate the dynamics via ode45, the interpolated controls always caused the states to diverge from the DIDO solution as indicated in Figure 6.109.

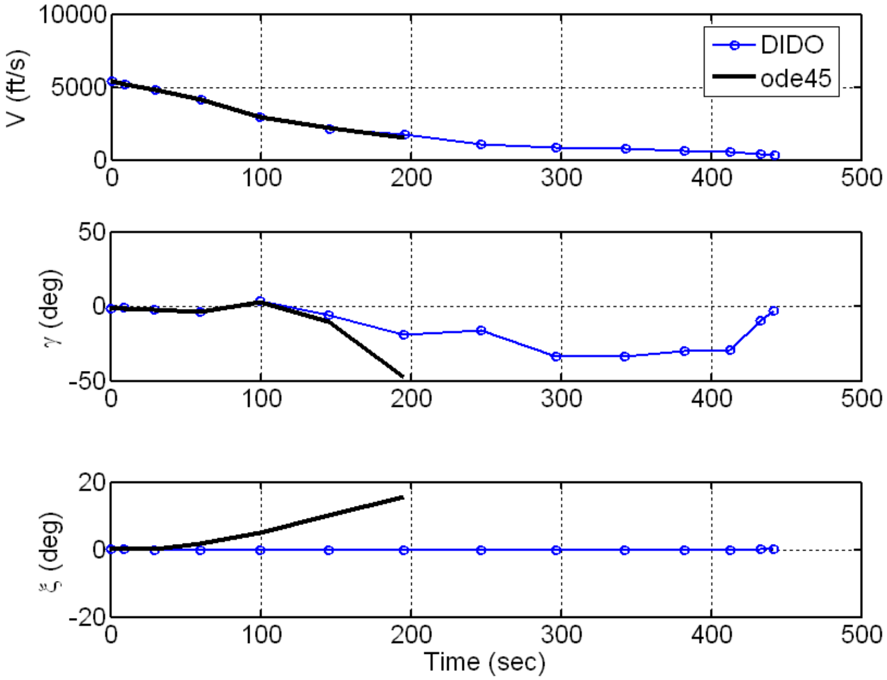


Figure 6.109 Example of 6-DOF Propagation Failure.

In some cases, this divergence corresponded to a warning in MATLAB:

> Warning: Failure at t=xxxxx. Unable to meet integration tolerances without reducing the step size below the smallest values allowed at time t.

Note that this same exact phenomenon plagued Carter's 6-DOF optimal combat maneuver work for his F-18 HARV and UCAV-X (possibly the X-45) models [195]. Although he did not investigate the cause of the problem, his explanation was based on software compatibility issues between DIDO and its use of third party software, the TOMLAB/SOL (SNOPT) toolbox. Using a previous version of TOMLAB/SOL, he had no problems with propagation for a similar Navion model. Although there was no conclusive evidence of this explanation, it is a possibility. Another more viable possibility may have to do with the type of interpolation and/or propagation being used.

Assuming that the selected interpolation scheme (cubic) does not cause the problem, two common versions of the MATLAB Runge-Kutta propagator were used to propagate a "cubic" interpolated control history. Those compared were the 4/5th-order "ode45" for non-stiff differential equations and the variable order, "ode15s" for stiff differential equations. Figure 6.110 to Figure 6.113 compare the ode45 results using adaptive and fixed step size. Note that ode15s gave approximately the same results as ode45; only sometimes did it propagate a little longer before diverging (not shown). As illustrated, a fixed step size improves the propagation, but still indicates signs of divergence.

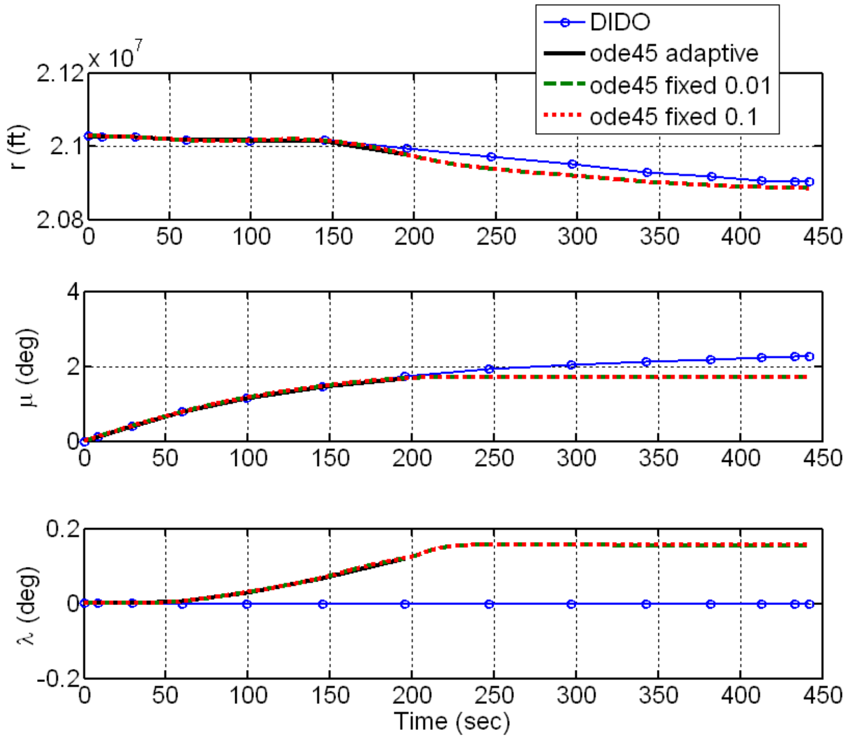


Figure 6.110 State Propagation using “ode45” (Position).

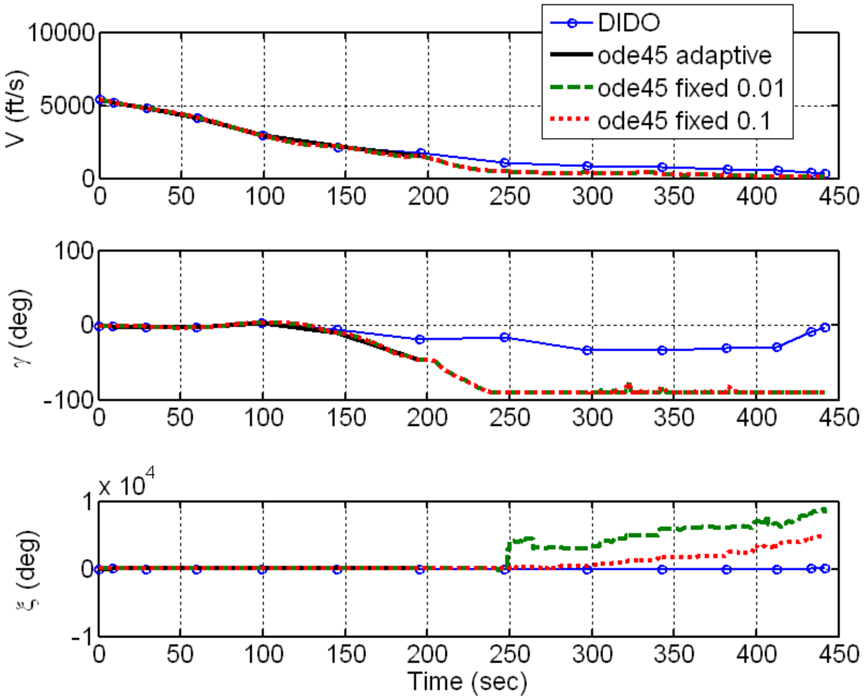


Figure 6.111 State Propagation using “ode45” (Velocity).

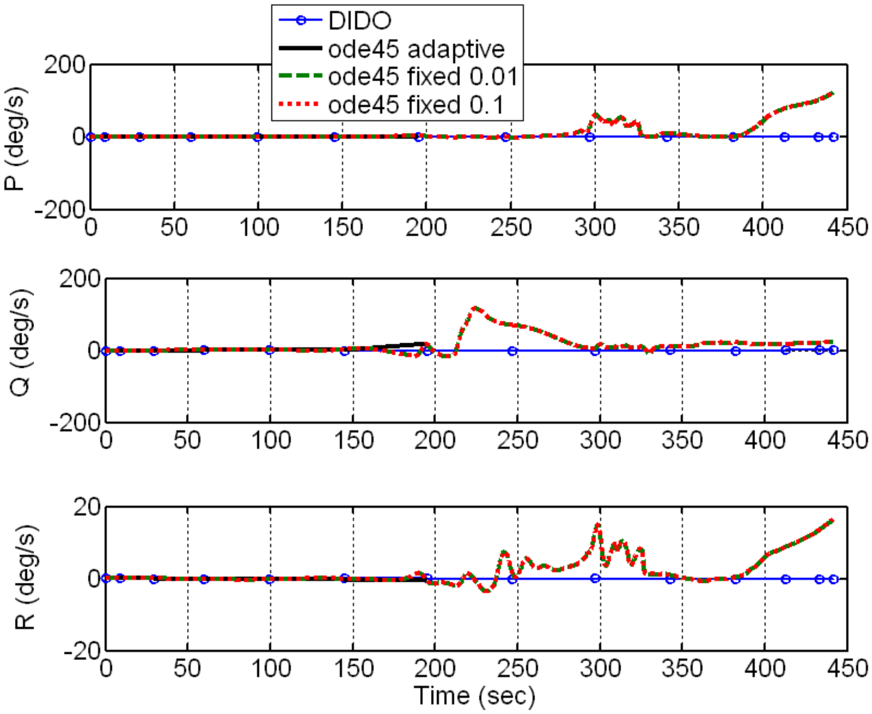


Figure 6.112 State Propagation using “ode45” (Body Rates).

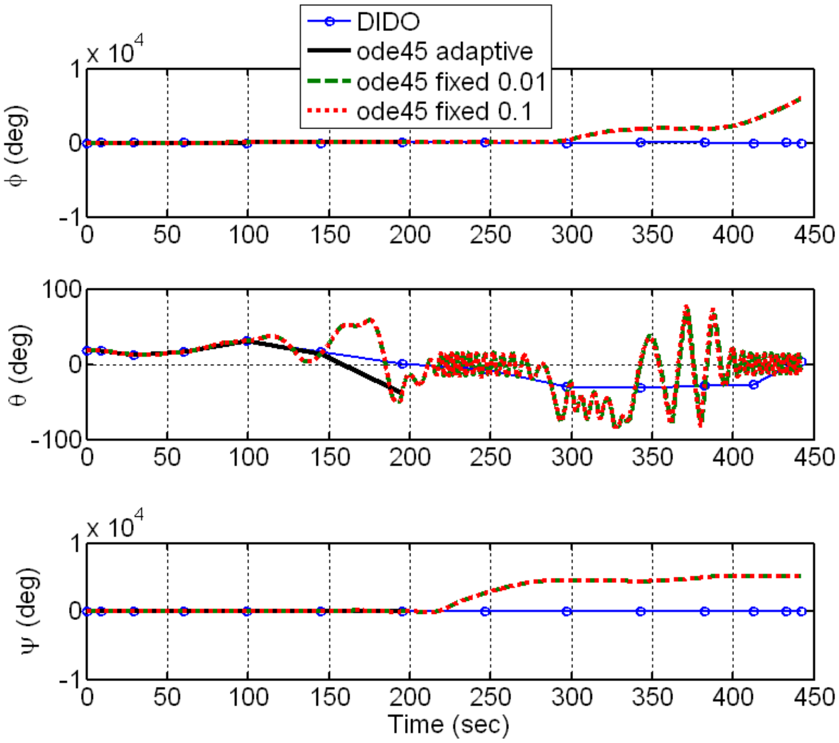


Figure 6.113 State Propagation using “ode45” (Euler Angles).

Note that it was originally thought that the interpolation scheme had little affect on the propagation results, but as exposed by Lewis, who experienced a similar problem in his work involving optimal path planning for unmanned ground vehicles, the system response to the interpolated controls is very sensitive [178]. With this, it was prudent to take a second look at the way MATLAB interpolates the RLV controls of this problem using various MATLAB interpolation schemes (e.g., linear, spline, cubic). A shown in Figure 6.114 to Figure 6.116, there are some variations in the interpolated controls, especially for the “spline” interpolant that has a tendency to overshoot. Lewis also associated the sensitivity of the propagation errors to the number of nodes used to generate the optimal controls. Obviously, the sensitivity of the interpolation and propagation methods is a concern, especially for closed-loop implementation that requires use of Runge-Kutta-type propagation; hence, this issue warrants future investigation.

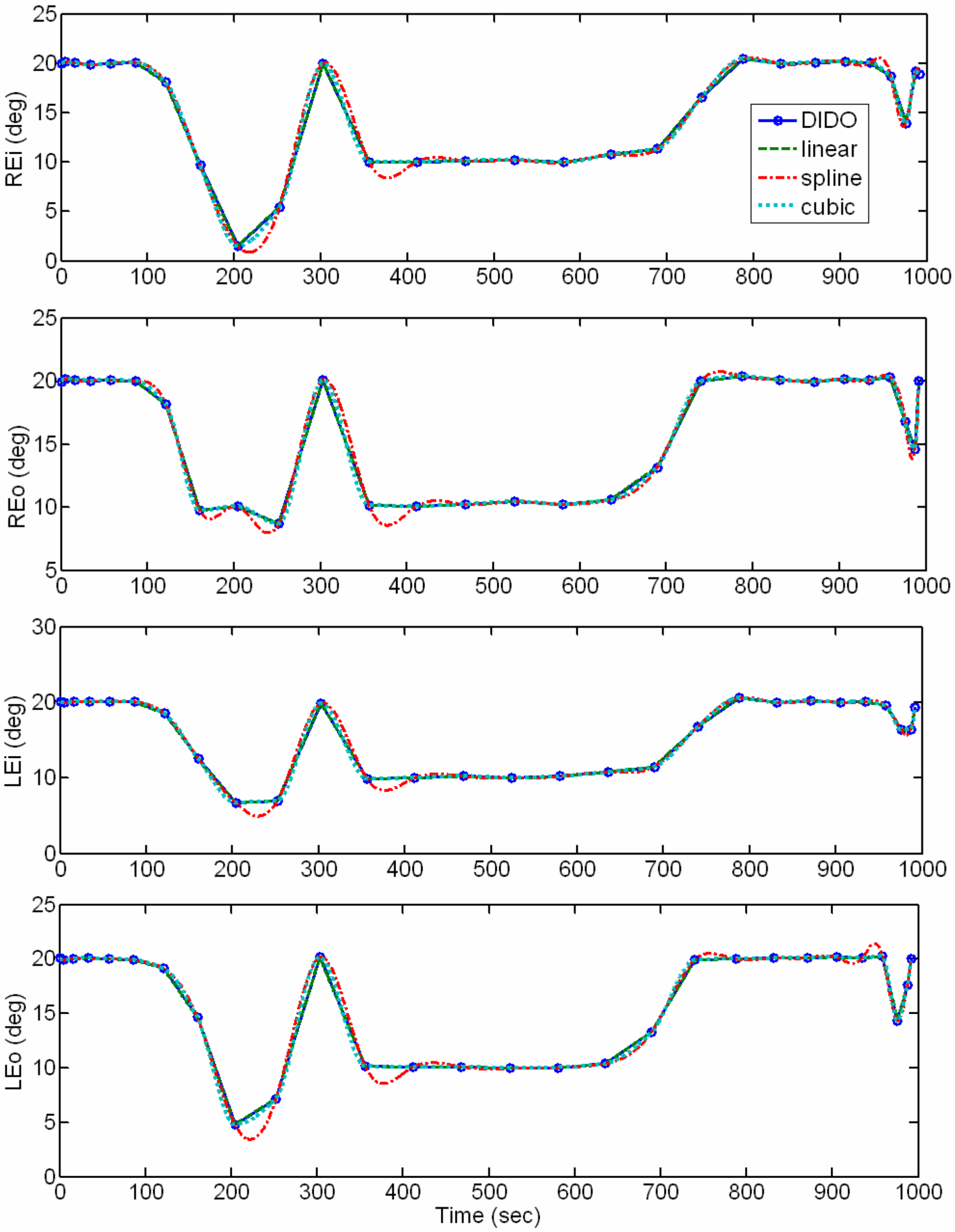


Figure 6.114 Interpolated Elevon Controls using Various MATLAB Schemes.

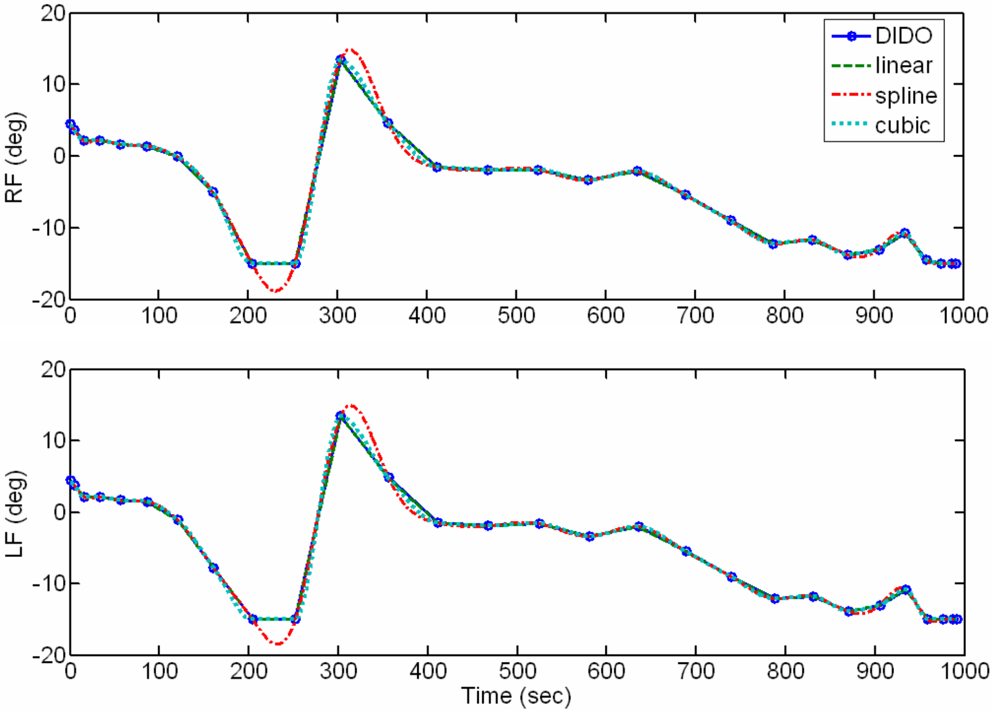


Figure 6.115 Interpolated Flap Controls using Various MATLAB Schemes.

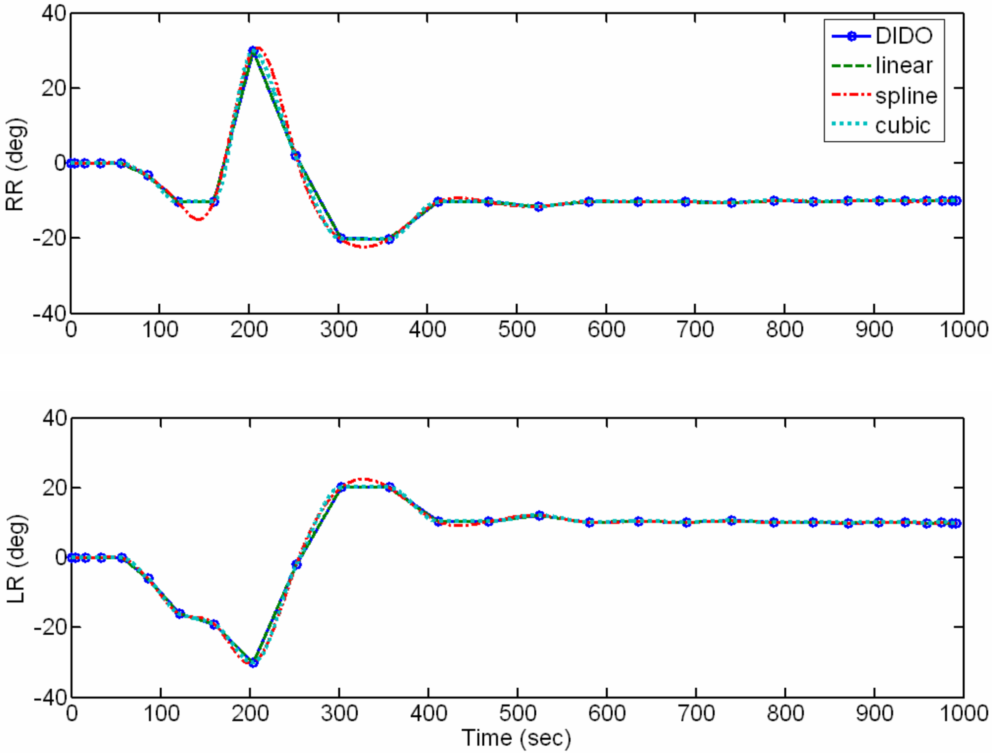


Figure 6.116 Interpolated Rudder Controls using Various MATLAB Schemes.

This work does not intend to substantiate Carter's claim about software compatibility, Lewis' claim about sensitivities as a result of grid size, or further investigate propagation methods; however, due to the lack of positive results from the standard feasibility test, an alternate confirmation of feasibility is required.

The most obvious approach is to verify that the control solution is physically realizable, i.e., realistic in the sense that the controls could be used to steer the vehicle. The requirement for this is primarily dependent on control rates since actuator operation is constrained within certain rate limits. For the X-33 control surface deflection rates, the elevons and rudders are limited to 30 deg/s whereas the body flaps are limited to 20 deg/s. Reviewing the 6-DOF control solution it is obvious that the rates are small, but to confirm this, the approximate rates were calculated and plotted in Figure 6.117. As shown, the rates are not even close the vehicle limits, but are all within +/- 1.0 deg/sec.

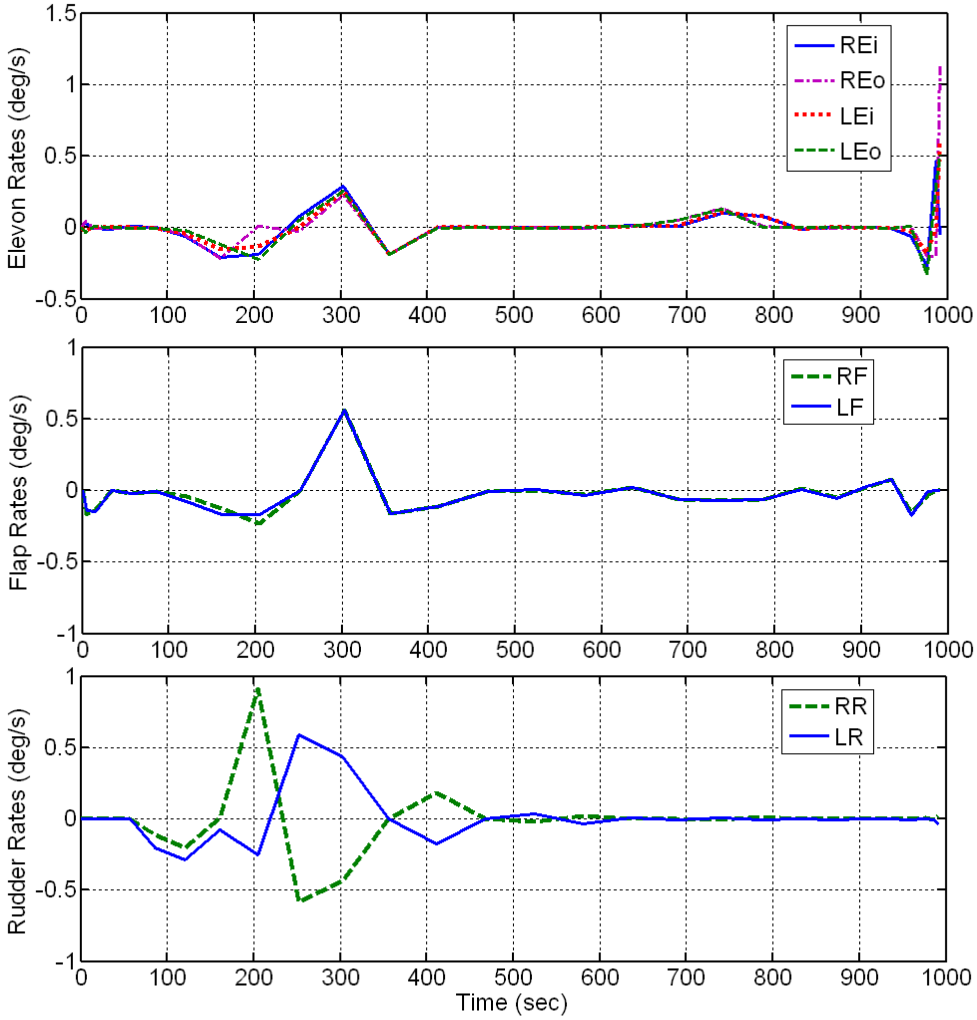


Figure 6.117 Example of Small Control Rates for 6-DOF Max DR Solution.

Any issues with exceeding rate limits can potentially be avoided by using the alternate model formulation that employs pseudo-controls; hence, allowing direct constraints on the rate limits. This also has the advantage of smoothing the control signal as given by the 3-DOF example in Chapter IV.

Also, for independent verification, the 6-DOF high-fidelity (HiFi) optimal solution is compared to the results of the 3-DOF commanded, 6-DOF tracking simulation. Since the inner-loop controller of the simulation generates 6-DOF control histories using piecewise linear programming (PLP) in the control allocator, it is conceivable that the controls will not be exact. Additionally, the results generated in that example were based on a 3-DOF model using slightly different assumptions and conditions. For example, the 3-DOF model used to generate the optimal guidance commands was based on a flat-earth model. Nonetheless, the trends of the control-time histories along with the generated forces and moments can provide a generalization of the solution feasibility. As such, Figure 6.118 to Figure 6.126 show the controls and the corresponding forces and moments from the two different solutions.

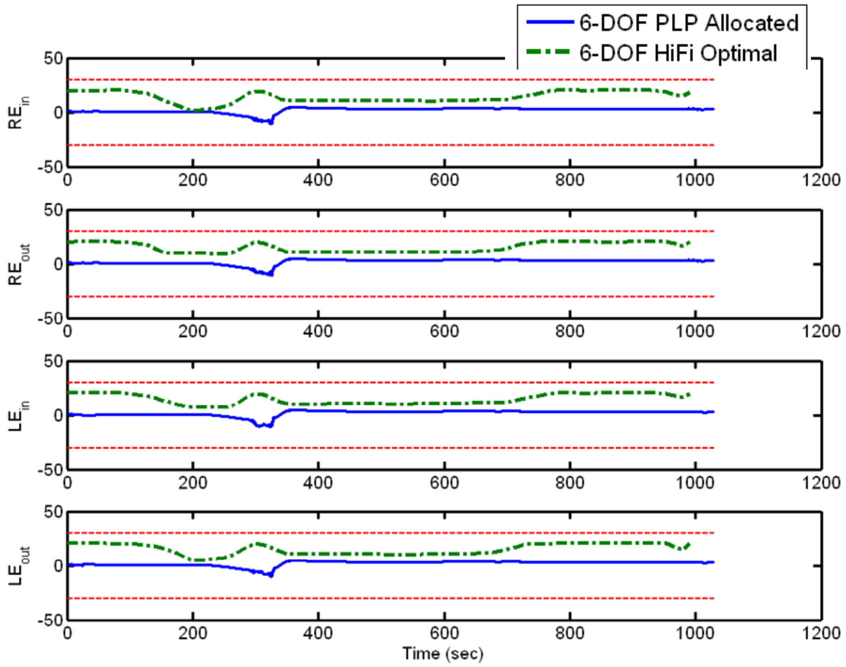


Figure 6.118 6-DOF HiFi Optimal vs. PLP Allocated Elevon Controls.

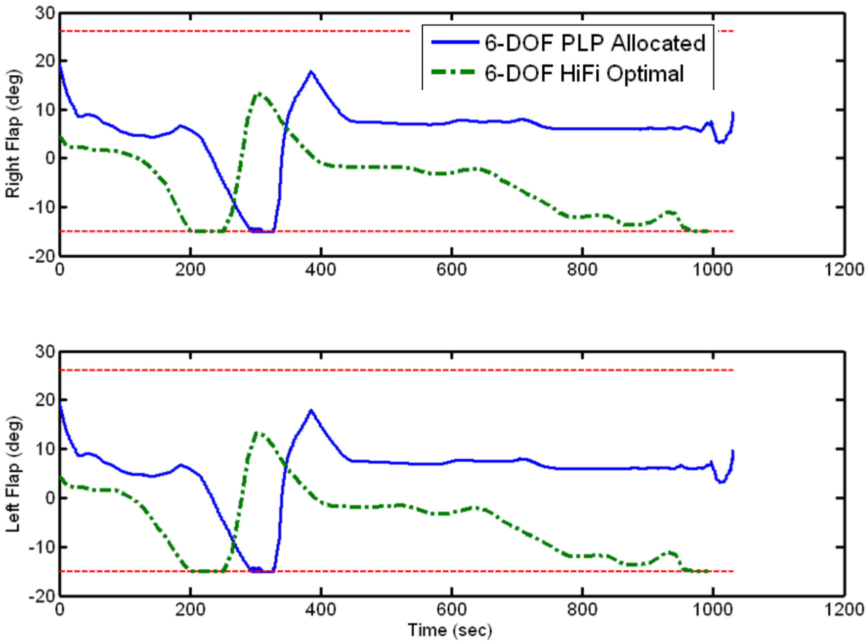


Figure 6.119 6-DOF HiFi Optimal vs. PLP Allocated Flap Controls.

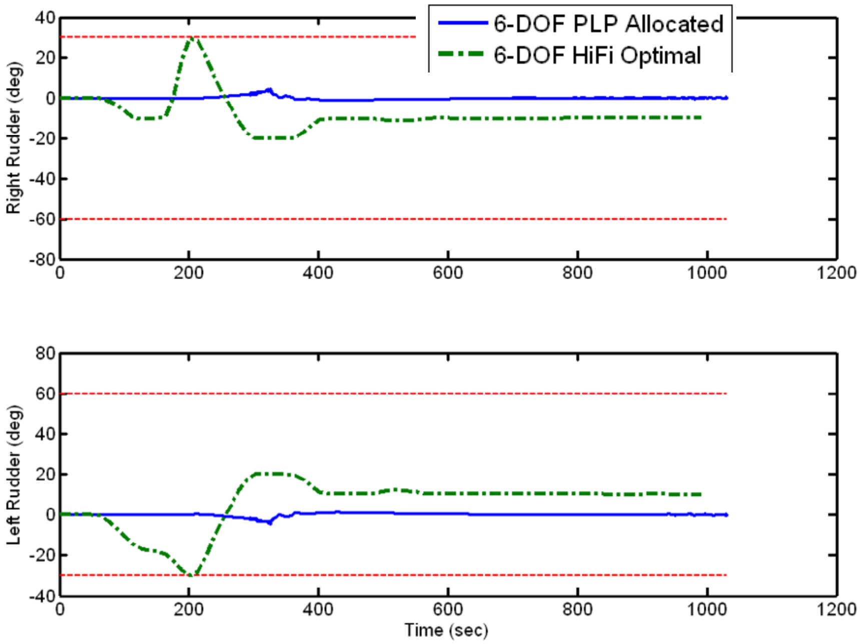


Figure 6.120 6-DOF HiFi Optimal vs. PLP Allocated Rudder Controls.

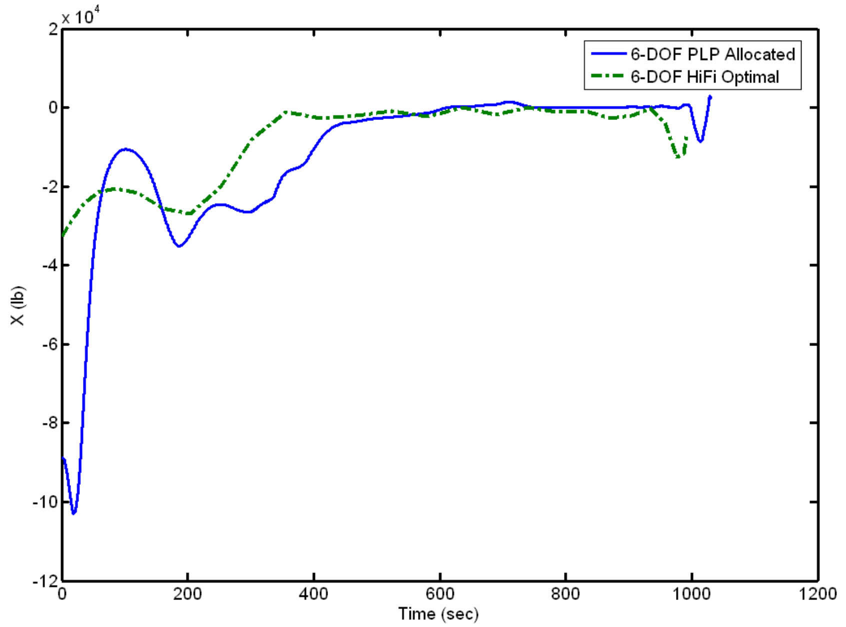


Figure 6.121 6-DOF HiFi Optimal vs. PLP Allocated Axial Force.

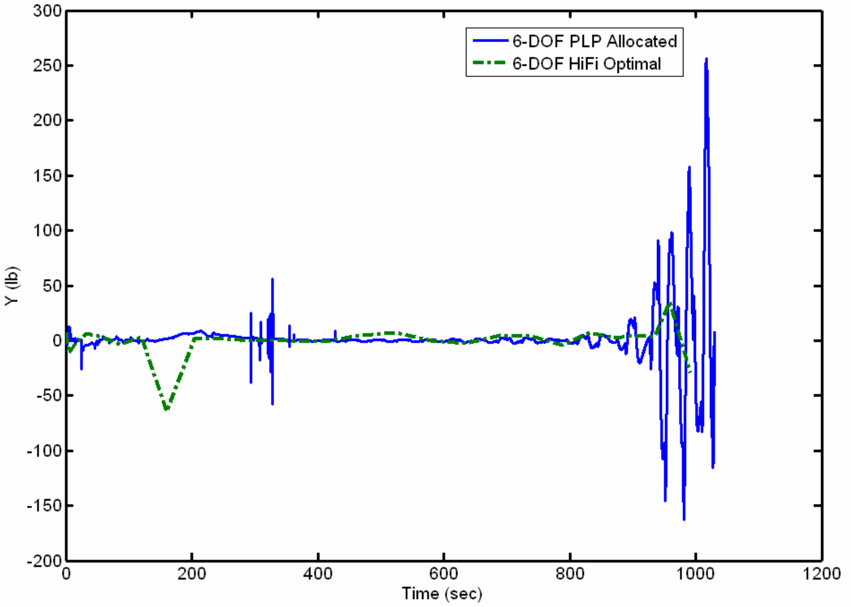


Figure 6.122 6-DOF HiFi Optimal vs. PLP Allocated Side Force.

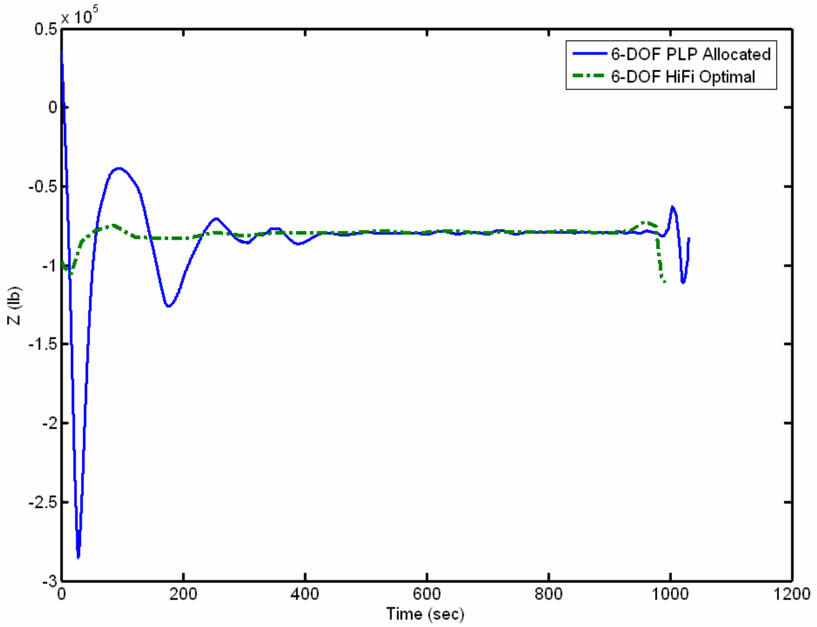


Figure 6.123 6-DOF HiFi Optimal vs. PLP Allocated Normal Force.

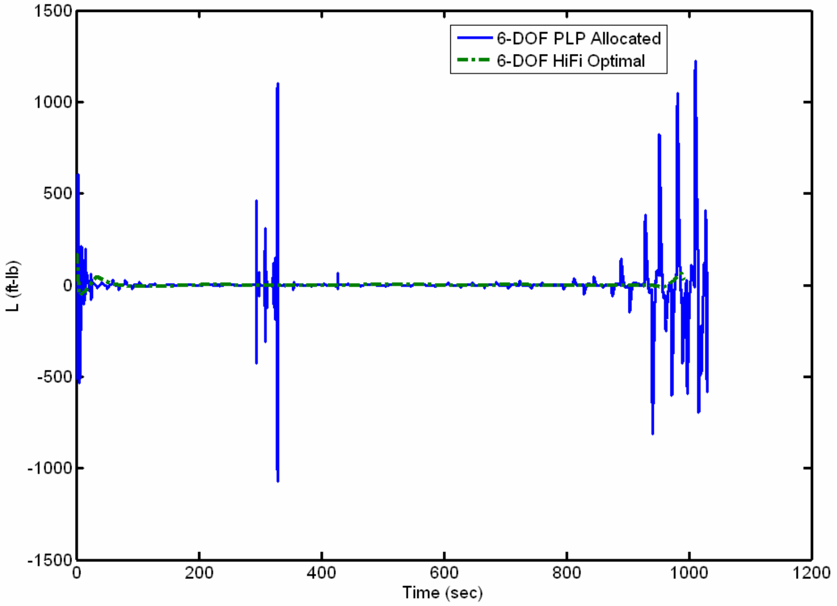


Figure 6.124 6-DOF HiFi Optimal vs. PLP Allocated Rolling Moment.

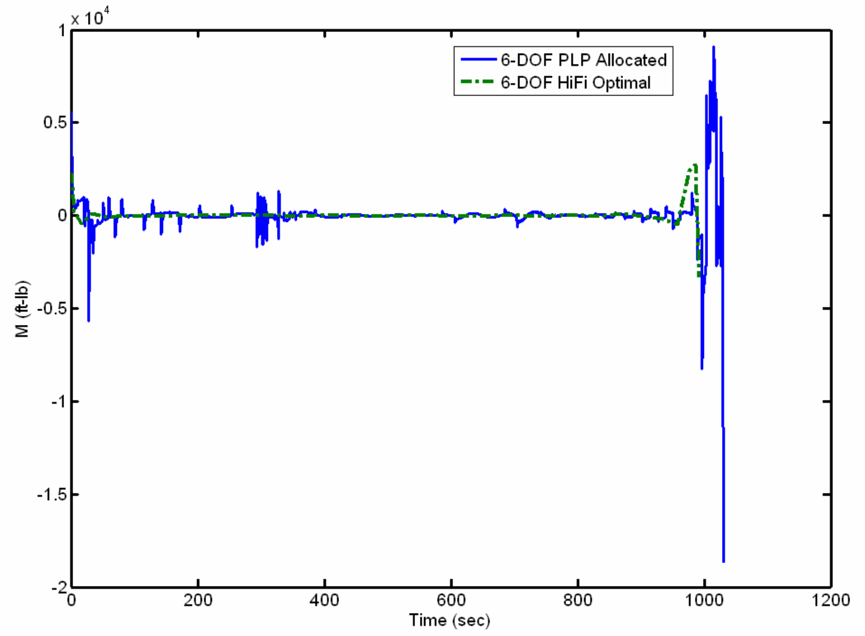


Figure 6.125 6-DOF HiFi Optimal vs. PLP Allocated Pitching Moment.

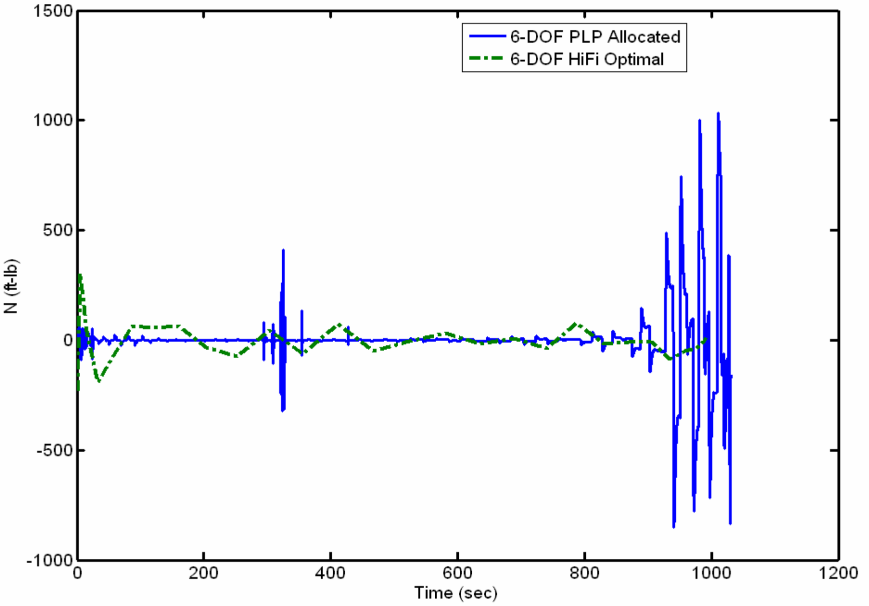


Figure 6.126 6-DOF HiFi Optimal vs. PLP Allocated Yawing Moment.

Note that since the 6-DOF HiFi optimal controls and the PLP allocated controls are generated using different models at slightly different initial conditions, only generalizations can be made about the trends. As seen in Figure 6.118, the trends for each of the elevon control histories are different in that the optimal deflections remain around 20 deg whereas the PLP allocated deflections remain around 0 deg and then switch to about 5 deg at approximately 325 sec. For a period of 325 to 500 sec, the optimal deflections drop to about 10 deg. This may be indicative of a different equilibrium point for the elevons. Figure 6.119 shows a very similar trend in flap control histories. Despite the apparent shift, the negative flap deflection magnitudes are approximately the same. An interesting result for the optimal deflections is that they gradually deflect downwards to their limits of around -15 deg whereas the PLP allocated flaps remain constant at about 10 deg. This downward deflection implies a nose-down pitching moment; however, at the same time the elevons are counter-reacting this as indicated by their upward deflections. According to the pitching moment history in Figure 6.125, the net effect is equivalent. Another striking difference is the rudder control histories. There is a large deflection for the optimal rudders; however, the right and left rudders appear to counter one another as indicated in Figure 6.120. As for the forces and moments, they show relatively similar trends. The largest difference occurs in the initial normal force. As shown in Figure 6.123, the optimal force is substantially lower than the PLP allocated force. This is likely related to the initial climb maneuver of the simulation trajectory as a consequence of a higher initial velocity.

Perhaps a more qualitative way to confirm both feasibility and optimality is by applying Bellman's Principle once again. In this case, the idea is to string a series of Bellman segments together to reconstruct the optimal solution.

9. Anti-Aliasing Bellman Segments

Solving the RLV reentry problem has the underlying difficulty of optimizing controls subject to both slow translational and fast rotational dynamics. This section applies the anti-aliasing bellman (a^2B) algorithm described in Chapter II in an attempt to capture the high-frequency effects that the previous 6-DOF solution may have missed. This will also serve as yet another way to confirm both feasibility and optimality of the solution. Note that this implementation is equivalent to the closed-loop method applied on the 3-DOF problem except that the clock-time is fixed. At the time of this writing, the large computational time for the 6-DOF optimization problem precluded use of the PS-feedback method. A direct benefit of this method is that a high-node solution is not required up front. Figure 6.127 to Figure 6.129 show the Bellman-sequenced control histories with the resulting Bellman-sequenced position histories, Figure 6.130 and Figure 6.131.

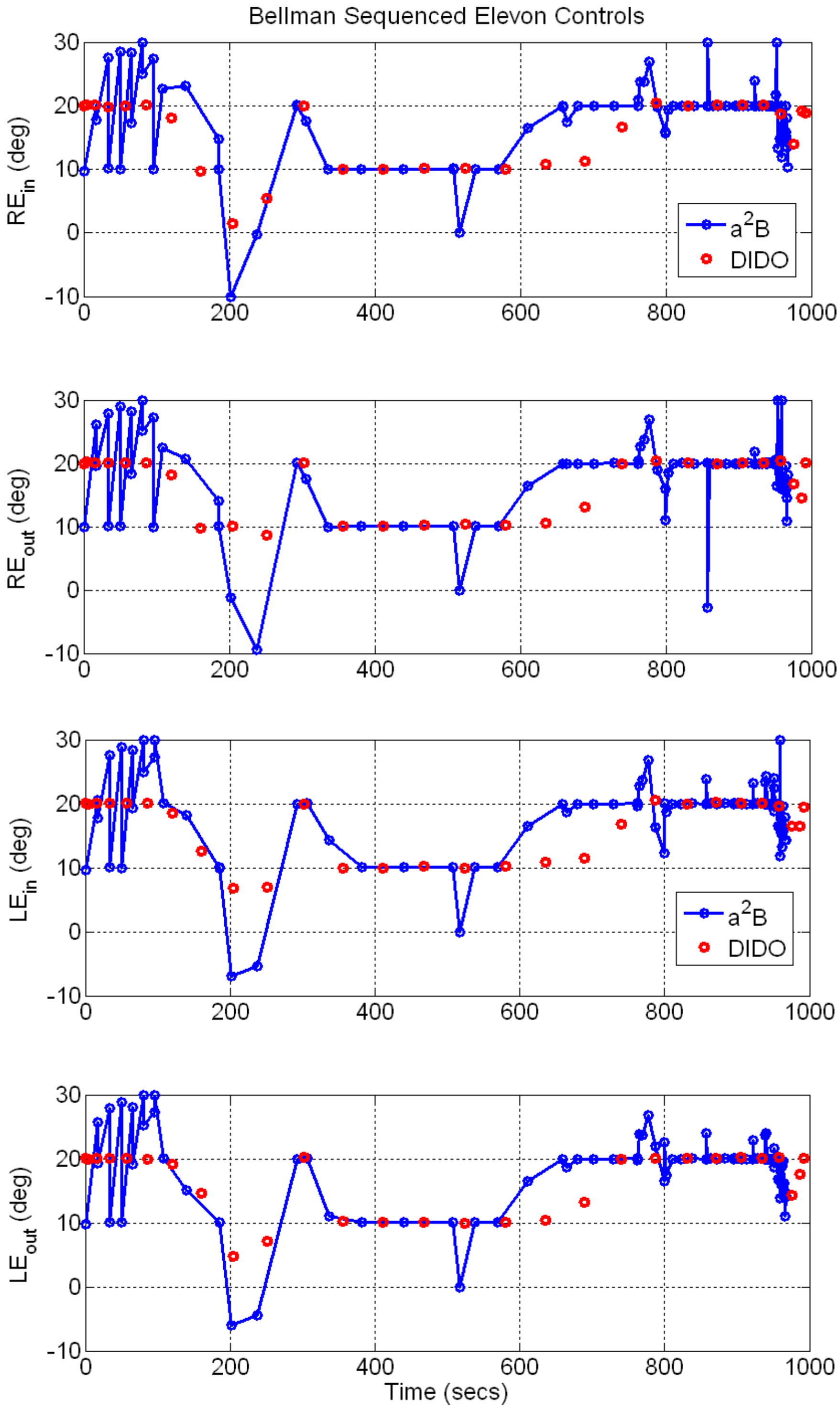


Figure 6.127 Bellman Sequenced Elevon Controls

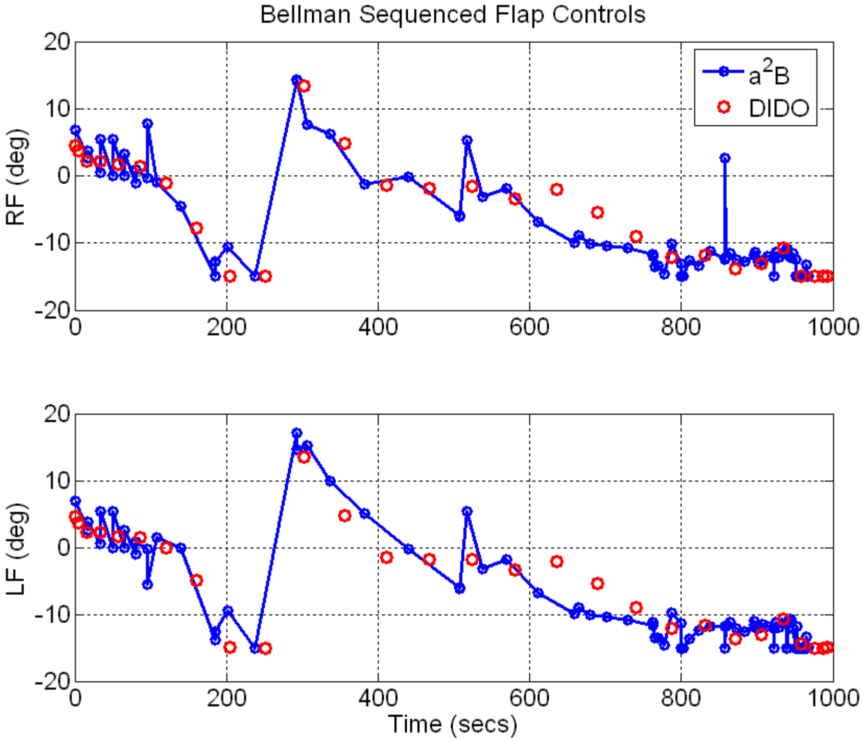


Figure 6.128 Bellman Sequenced Flap Controls.

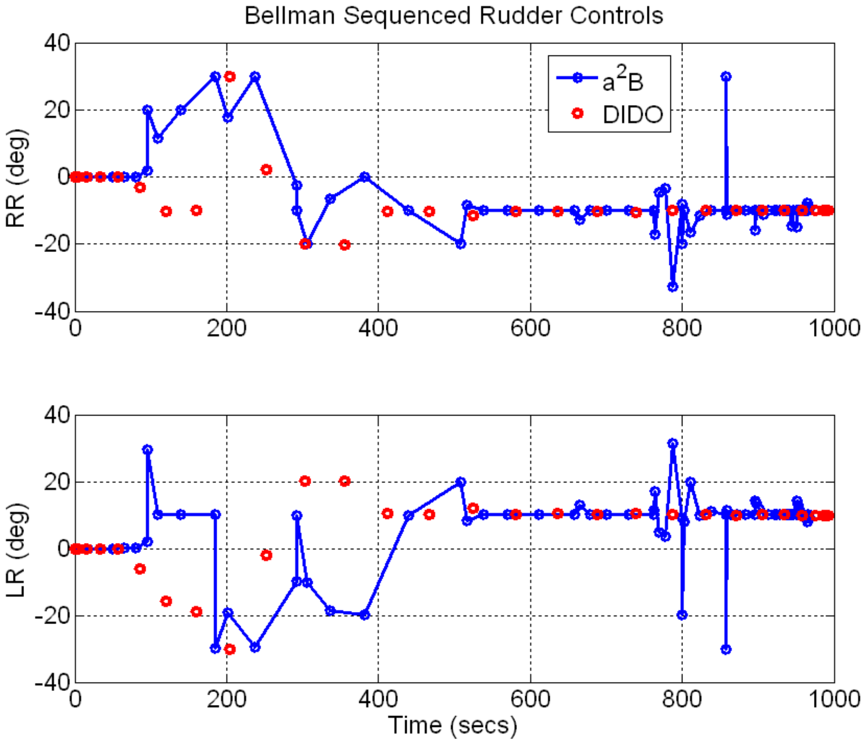


Figure 6.129 Bellman Sequenced Rudder Controls.

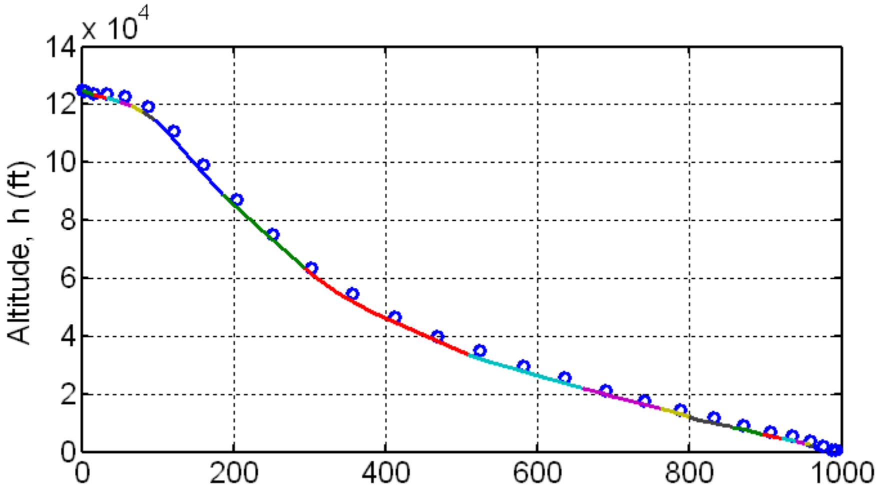


Figure 6.130 Bellman Sequenced Altitude.

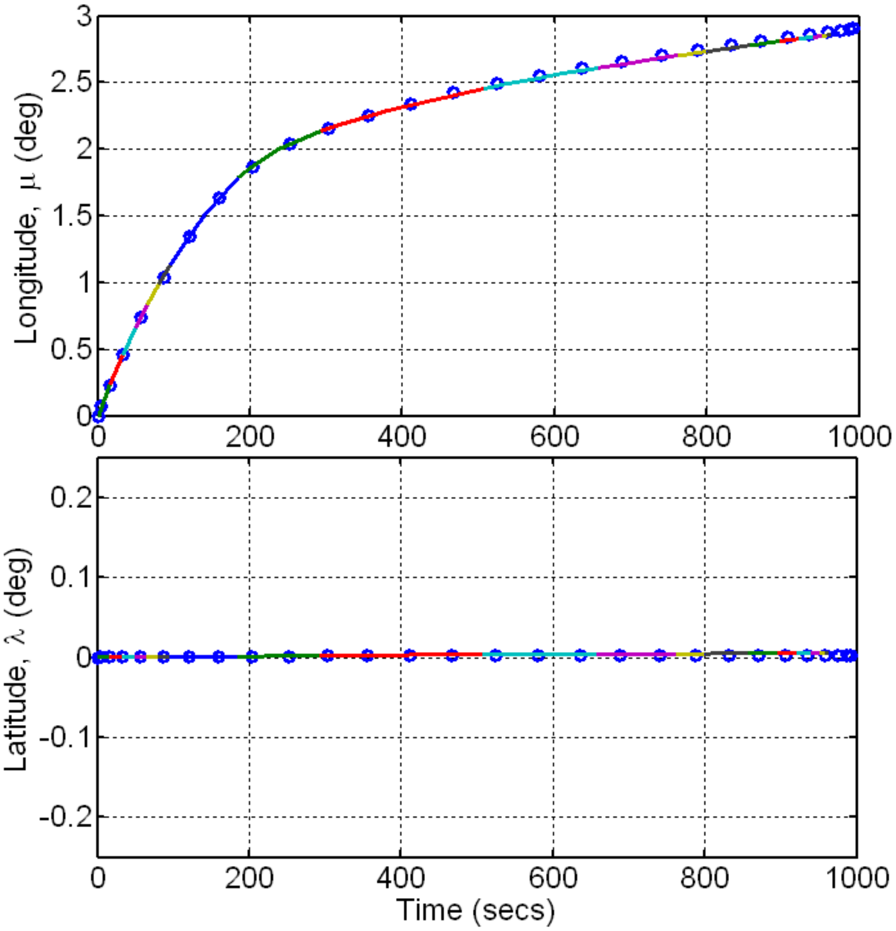


Figure 6.131 Bellman Sequenced Trajectory (Position).

As shown, using the a^2B-algorithm appears to capture more of the high-frequency effects from the rotational dynamics. An interesting result here is the presence of extremely rapid oscillations during the first 100 seconds for the elevons and the flaps. This may be an indication of an un-trimmed initial condition. Recall that this problem formulation specified the initial conditions. An appealing test would be to let all of the state variables except position and altitude be free and see if the solution still has these oscillations in the beginning. If not, then the computed optimal controls for the initial condition are likely the trimmed values.

Since portions of the linearly interpolated Bellman segments are far from the DIDO nodes, this is an indication that even larger oscillation may exists that were not anti-aliased. These Bellman sequenced control histories were generated using 20 segments with each successive initial condition separated by two, four, or six nodes with a combined equivalence of 92 nodes. To anti-alias the signal even more, a higher number of segments is required. From the appearance of these high-frequency oscillations this problem warrants the use of the alternate problem formulation using pseudo-controls. This will help smooth the controls and allow constraints to be placed directly on the control rates.

10. Footprint Generation

The 6-DOF footprint generation used both Problems SS and MS. Consistent with the 3-DOF footprint scenarios, the constraints on states, controls, paths and final endpoint conditions remained the same, but initial conditions were adjusted accordingly.

Initial Conditions:

$$\begin{array}{llll} \mathrm{h(t_0)} = 167{,}323 \text{ ft} & \mathrm{V(t_0)} = 8530 \text{ ft/s} & \mathrm{p}(t_0) = 0\,\text{deg/s} & \phi(t_0) = 0\,\text{deg} \\ \mu(\mathrm{t_0}) = \text{-85 deg} & \gamma(t_0) = \text{-1.5 deg} & \mathrm{q}(t_0) = 0\,\text{deg/s} & \theta(t_0) = 28.5 \text{ deg} \\ \lambda(\mathrm{t_0}) = 26 \text{ deg} & \xi(t_0) = 0 \text{ deg} & \mathrm{r}(t_0) = 0\,\text{deg/s} & \psi(t_0) = 0 \text{ deg} \end{array} \tag{5.73}$$

with the following initial and final guesses (linear interpolation between two points)

$$
\begin{aligned}
&h_0 = 167{,}323 \text{ ft} && h_f = 500 \text{ ft}\\
&\mu_0 = -85 \text{ deg} && \mu_f = -78 \text{ deg}\\
&\lambda_0 = 26 \text{deg} && \lambda_f = 26 \text{deg}\\
&V_0 = 8530 \text{ ft/s} && V_f = 335.13 \text{ ft/s}\\
&\gamma_0 = \text{-}1.5 \text{ deg} && \gamma_f = -3 \text{ deg}\\
&\xi_0 = 0 \text{ deg} && \xi_f = 0 \text{ deg}\\
&p_0 = 0 \text{deg/s} && p_f = 0 \text{deg/s}\\
&q_0 = 0 \text{deg/s} && q_f = 0 \text{deg/s}\\
&r_0 = 0 \text{deg/s} && r_f = 0 \text{deg/s}\\
&\phi_0 = 0 \text{deg} && \phi_f = 0 \text{deg}\\
&\theta_0 = 28.5 \text{ deg} && \theta_f = 0 \text{ deg}\\
&\psi_0 = 0 \text{ deg} && \psi_f = 0 \text{ deg}
\end{aligned}
\tag{5.74}
$$

$$
\begin{aligned}
&\delta_{\text{REi}_0} = \text{-}0.328 \text{ deg} && \delta_{\text{REi}_f} = 0 \text{ deg}\\
&\delta_{\text{REo}_0} = \text{-}0.328 \text{deg} && \delta_{\text{REo}_f} = 0 \text{deg}\\
&\delta_{\text{LEi}_0} = \text{-}0.328 \text{deg} && \delta_{\text{LEi}_f} = 0 \text{deg}\\
&\delta_{\text{LEo}_0} = \text{-}0.328 \text{ deg} && \delta_{\text{LEo}_f} = 0 \text{ deg}\\
&\delta_{\text{RF}_0} = 9.17 \text{ deg} && \delta_{\text{RF}_f} = 0 \text{ deg}\\
&\delta_{\text{LF}_0} = 9.17 \text{deg} && \delta_{\text{LF}_f} = 0 \text{deg}\\
&\delta_{\text{RR}_0} = 0 \text{deg} && \delta_{\text{RR}_f} = 0 \text{deg}\\
&\delta_{\text{LR}_0} = 0 \text{deg} && \delta_{\text{LR}_f} = 0 \text{deg}
\end{aligned}
\tag{5.75}
$$

The process of generating footprints for the 6-DOF Problems (SS and MS) presented some interesting results. First, when using approximate flight angle relationships there are errors that cannot be ignored, especially for large-angle maneuvers involving large bank and roll angles such as the case in maximum crossrange trajectories. The maximum downrange solutions were as expected, but the maximum crossrange solutions revealed some important discrepancies. As illustrated in Figure 6.132, for a maximum downrange case, there is negligible error between the approximate and exact flight angle relations. On the other hand, for a maximum crossrange case, Figure 6.133 shows substantial error between the two. Additional variations of angles confirmed the commonly accepted “rule-of-thumb” that the approximate relations are only valid for small angles on the order of 10 deg or less (see Figure 6.134). The effects of this may not be apparent until trying to run crossrange simulations in which infeasible trajectories may result.

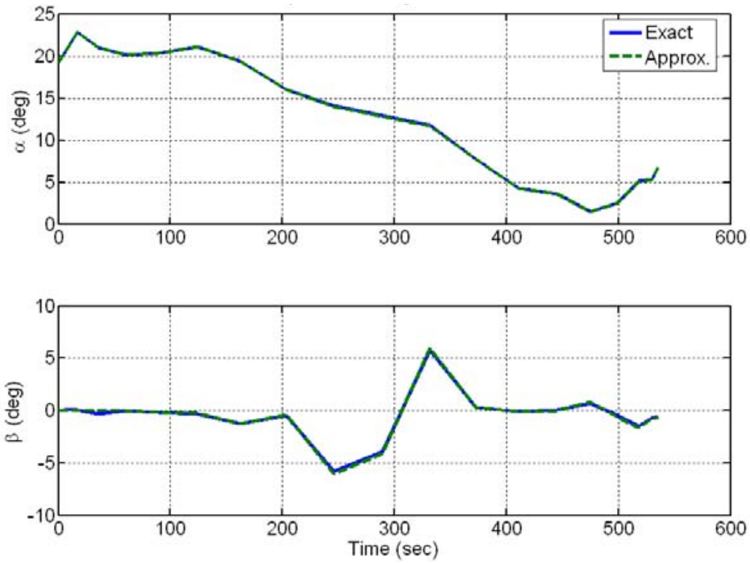


Figure 6.132 Comparing Aero Angles for Max DR (DIDO Trajectory).

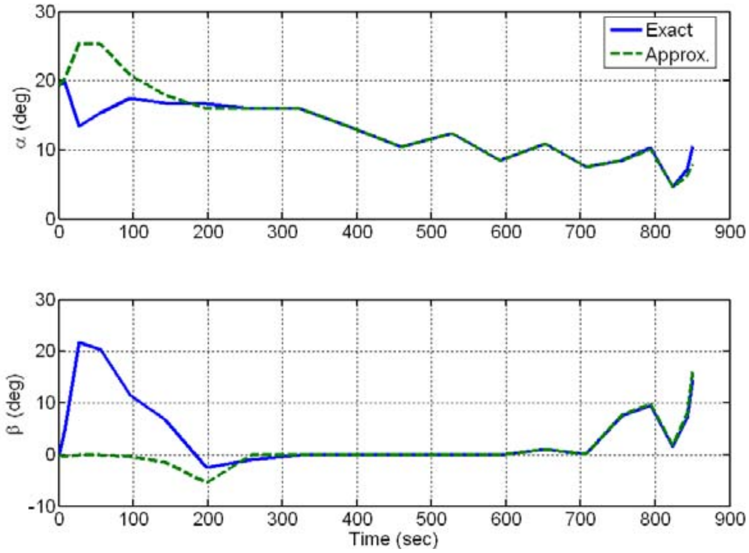


Figure 6.133 Comparing Aero Angles for Max CR (DIDO Trajectory).

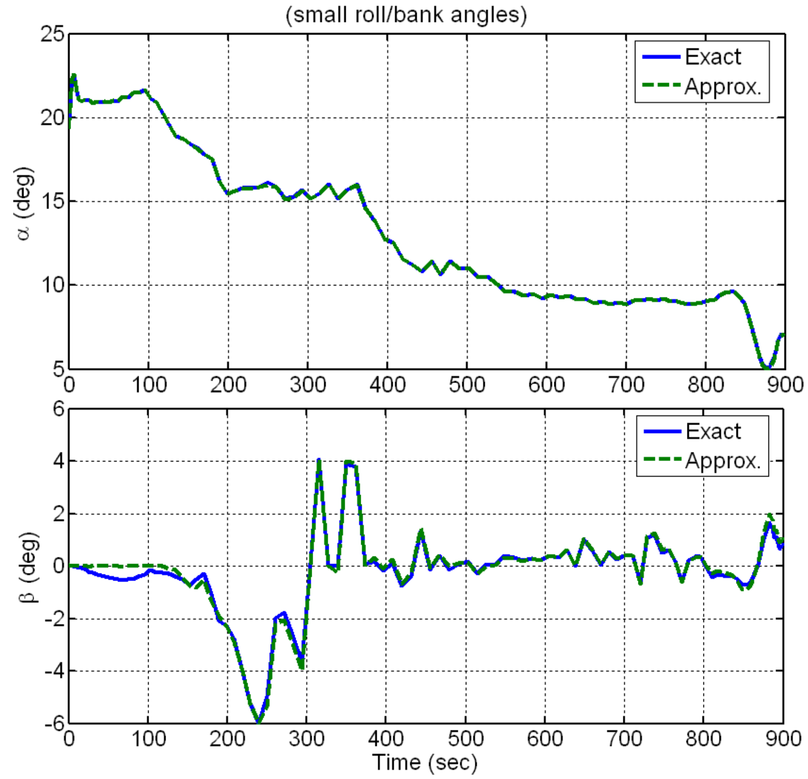


Figure 6.134 Comparing Aero Angles for Small Bank/Roll (< 10 Degrees).

Forcing zero sideslip allowed feasible crossrange trajectories since this did not allow any errors to negatively affect the lateral-directional mode. When using the exact angle relations and allowing nonzero sideslip angles in the problem formulation, even the maximum downrange case *occasionally* experienced unexpected results. As shown in Figure 6.135 and Figure 6.136, oscillations occur in the lateral-directional channels at approximately 307 seconds. This sudden sideslip/yaw and bank causes the flight path to have a rather abrupt turn off of the nominal, expected path for a maximum downrange case. Consequently, this maneuver degrades the downrange performance. Surprisingly, it was found in the literature that in addition to sideslip angle being "the most sensitive state to overall vehicle performance" [32], there is also a critical entry condition for the X-33 that occurs at about M=3.16 at an altitude of approximately 97,167 ft (near TAEM) associated with a "critical, unstable lateral-directional point" [45]. A closer look at the DIDO solution revealed that this sudden maneuver corresponds to M=3.22~2.37 at an altitude of 109,687~98,090 ft. After 307 sec, the sideslip angle continues to oscillate. The results also indicate active rudders during this time which could be a sign that they are trying to compensate for the oscillatory behavior or this action may be contributing to the oscillations. The concern is on rudder effectiveness during this time. Recall, that rudders may not be effective until $M \leq 2.5$ and the onset of this "critical condition" starts at $M \approx 3.20$; therefore, the inclusion of this rudder effectiveness condition given by Eq. (5.49) may be required after all.

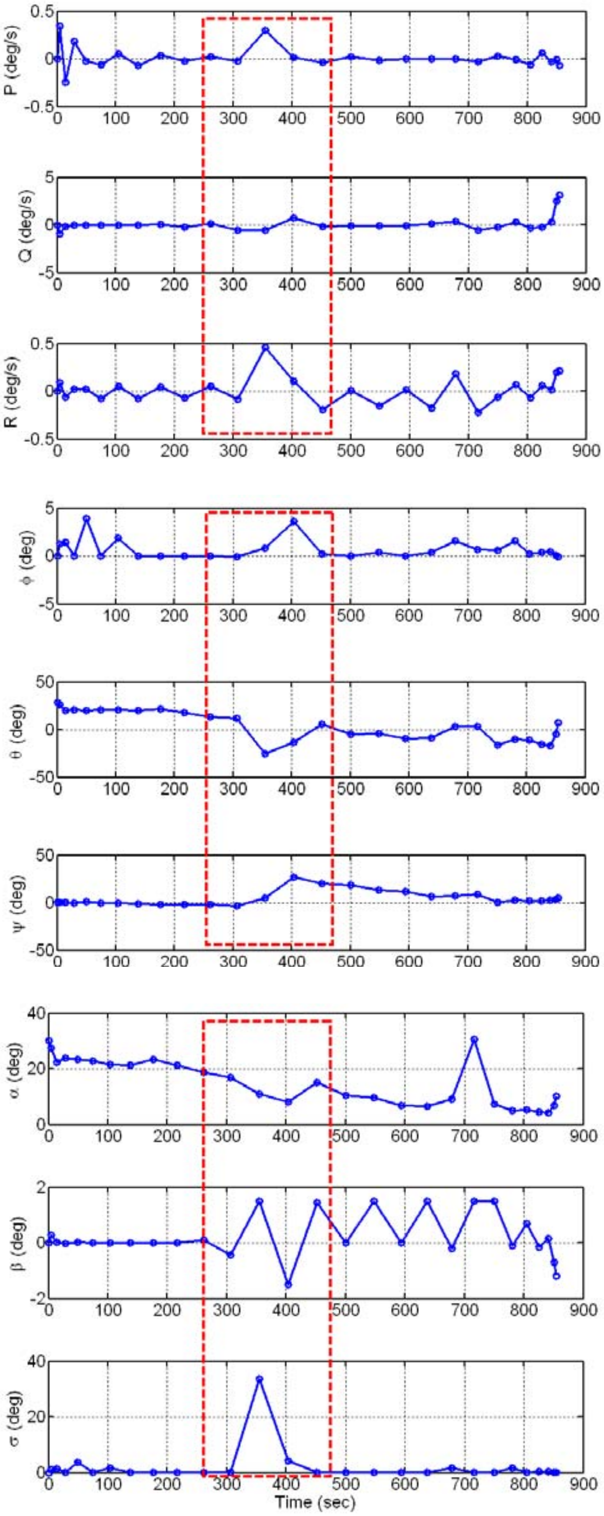


Figure 6.135 Example of Lateral-Directional Anomaly (States).

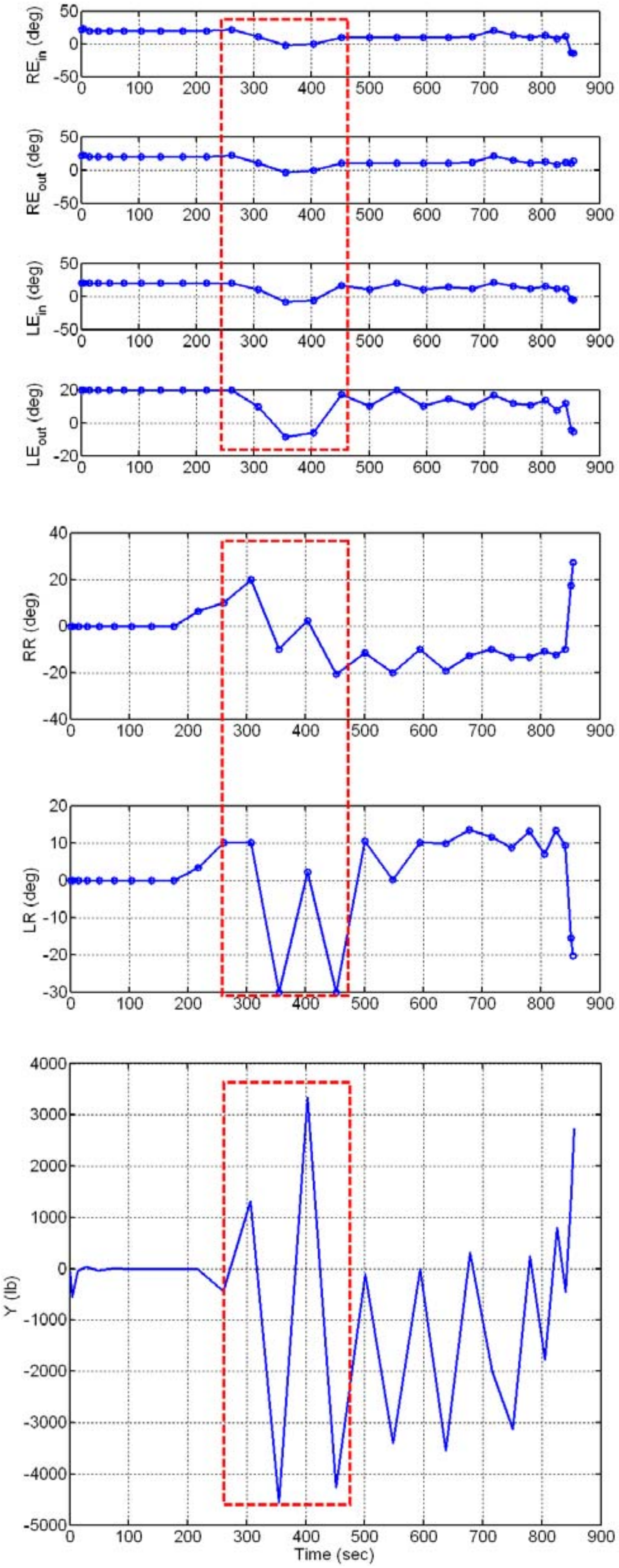


Figure 6.136 Example of Lateral-Directional Anomaly (Controls and Side Force).

Similar to the 3-DOF work, the 6-DOF model is used to generate polygonal footprint approximations indicating the maximum reachable set of feasible landing sites. Figure 6.137 shows footprints for the various 6-DOF cases compared to the 3-DOF footprint from Chapter IV.

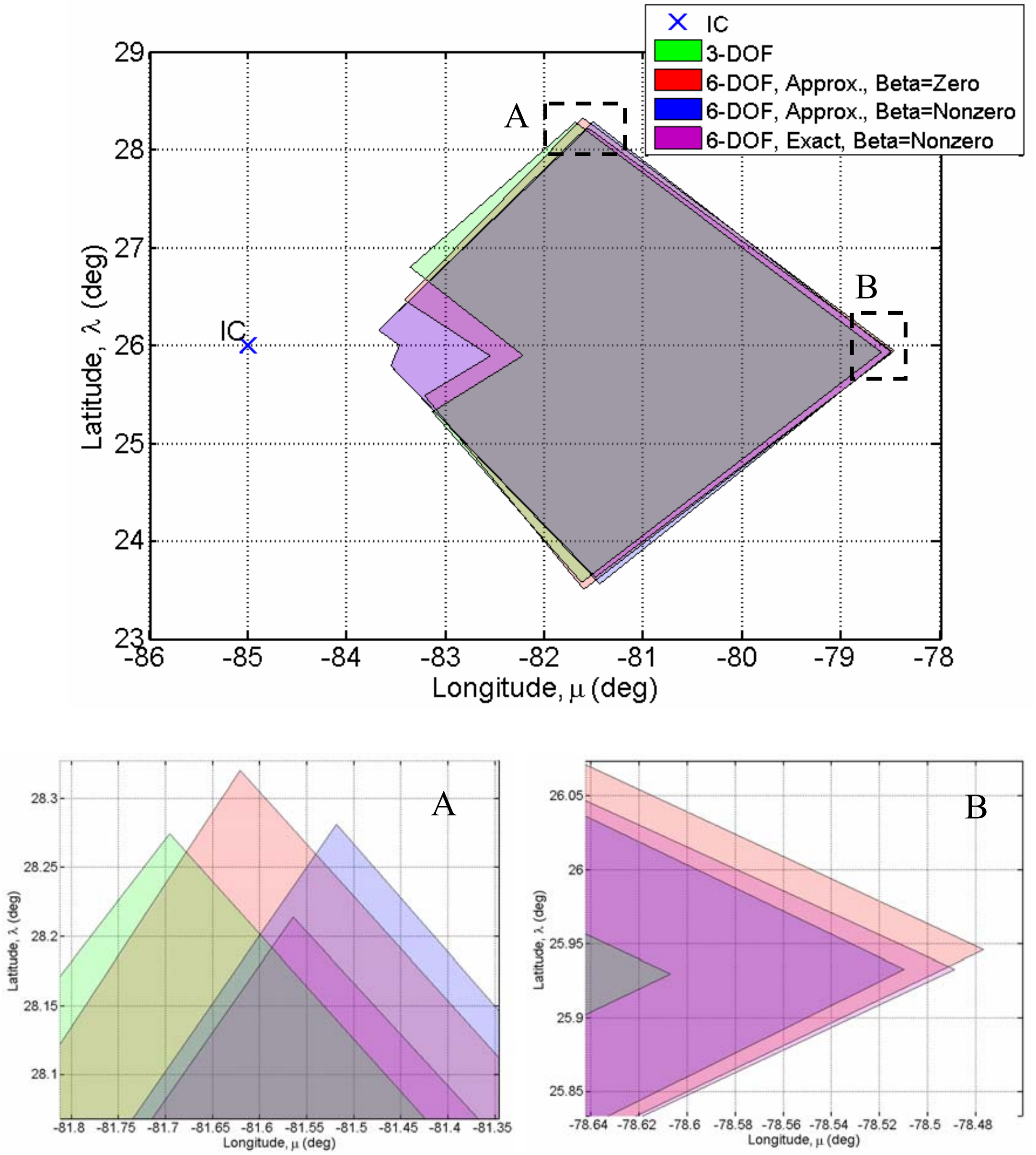


Figure 6.137 6-DOF Footprints vs. 3-DOF Footprint.

As shown, the overall trend between each footprint is similar. Comparing the extremal points shown in the inset plot “A” of Figure 6.137 indicates that the variance between x-coordinates is 0.0482 and y-coordinates is 0.0061. Comparing the extremal points shown in the inset plot “B” of Figure 6.137 indicates that the variance between x-coordinates is 0.0001 and y-coordinates is 0.0040. These variations are consistent with the previous 3-DOF versus 6-DOF differences on the order of 3 to 8 nm. Interestingly, the points for the nonzero sideslip cases both indicate smaller minimum ranges. This verifies the conjecture made about the possible performance degradation when forcing zero sideslip angles for low-speed flight. In this case, the footprints indicate that a nonzero sideslip angle improves the minimum downrange performance of the RLV.

11. Numerical Considerations

a. Computational Speed

Table 6.5 illustrates the effects that bootstrapping has on the CPU runtimes for Problem HF1. Following the example of Carter, the percentages listed in the table are a measure of how much faster the runs were completed as a result of bootstrapping from the respective number of nodes [195]. For example, the complete runtime to bootstrap from 20 to 80 nodes takes 60.7 sec (8.5 sec + 52.2 sec), whereas 80 nodes by itself takes 121.8 sec; therefore, 50% longer than the original combined bootstrapping time. In other words, the effect of bootstrapping in this case saves 50% in computational time.

Table 6.5 Effect of Bootstrapping on CPU Runtime (secs) for Problem HF1

		Bootstrapped # of Nodes					
	# Nodes	**0**	**20**	**40**	**60**	**80**	**100**
Original # of Nodes	**14**	4.2	6.2 122%	11.8 66%	30.5 51%	51.6 46%	59.3 13%
	20	8.5		12.6 86%	25.3 50%	52.2 50%	89.9 20%
	40	24.4	X		26 75%	51.8 63%	76.6 20%
	60	67.5	X	X		45.5 93%	?
	80	121.8	X	X	X		78.8 41%
	100	493	X	X	X	X	

X = omitted – “never, never bootstrap to a lower node value” [195]
? = not converged

Note that since Problem HF1 did not rely on the aero data, the CPU computation times were substantially faster. For example, Problem SS and MS CPU runtimes ranged from 300 to 1000 sec on average for 14 to 28 nodes. Compared to the times in Table 6.5, Problem SS and MS run significantly slower.

Computational speed is obviously a concern for the 6-DOF trajectory optimization, especially when using tabulated look-up data; however, this computation time can be reduced by at least a factor of 100 by optimizing the actual code, eliminating the Windows and MATLAB overhead etc. [151]. None of these computational enhancements were carried out because the purpose of the current work is to demonstrate the principles. Since use of aerodynamic data tables involves linear interpolation between flight condition parameters (e.g.,, between Mach numbers), one possible way to reduce the computational speed when using the tables is to use constant gradients between the linear interpolating points. This will reduce the number of gradient calculations; however, since the scope of this work required using DIDO software as a "black box," alterations to the source code was not an option.

b. Uniqueness of Solution

Although having redundant control surfaces provides a margin of safety, particularly in a scenario requiring reconfigurable control, it also introduces ambiguity into the control optimization algorithm. For the X-33, there are eight independently operated control surfaces. As presented, the control system is overactuated such that multiple control surface settings have the overall same effect. Although extensive research was not conducted in this area, one way to prevent this ambiguity in the controls is to define a secondary objective. Modifying the cost function to include additional objectives should theoretically provide a unique solution. Two suitable objectives that are simple to implement are that of minimum control effort and maximal preference allocation.

If the overall minimum control effort is desired, the cost function can take the form of the quadratic-control term of a Linear Quadratic Regulator (LQR) optimal control problem [80] such that

$$J(\cdot) = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}(t)^T \mathbf{R}\mathbf{u}(t)dt \tag{5.76}$$

where **R** is a real-symmetric, positive-definite weighting matrix. For the flight control problem of maximizing downrange (or crossrange), the cost function can be written as

$$\begin{aligned} J(\cdot) &= w_1\left(-\mu_f\right) + \frac{1}{2}\int_{t_0}^{t_f} \boldsymbol{\delta}(t)^T \mathbf{R}(t)\boldsymbol{\delta}(t)dt \\ &= w_1\left(-\mu_f\right) + \frac{1}{2}\int_{t_0}^{t_f} \left(\sum_{i=1}^{n} r_i \boldsymbol{\delta}_i(t)^2\right)dt \end{aligned} \tag{5.77}$$

where r_i are the non-negative weighting factors for each control (i=1..n). Assuming that the actuation of each control surface is equally important, the **R** matrix can be unity (i.e., r is constant for all i). Note that for max/min-crossrange, the final longitude (μ_f) in Eq. (5.77) is replaced by the final latitude (λ_f).

Another technique is to minimize control surface deflections by imposing a “maximal preference allocation” into the cost function [109]. This consists of specifying a “preferred” control surface position and often may be based on providing an extra margin of safety should the vehicle encounter an unexpected disturbance. For the allocation problem, the quadratic cost term can be expressed as

$$J(\cdot) = \sum_{i=1}^{n} (\delta_i - \delta_i^*)^T (\delta_i - \delta_i^*) \tag{5.78}$$

where n is the number of controls and δ^* is some “preferred” control setting. Previous efforts have used a similar approach whereby a preference vector based on trim conditions is provided in the control allocation problem [41]. This may be acceptable for nominal maximum-range flights, but for problems involving maneuvering flight, this approach may limit the vehicle’s performance by restricting its operations to equilibrium glide characteristics.

As a preliminary test case, the following cost function for the max-range case was implemented,

$$J(\cdot) = w_1\left(-\mu_f\right) + w_2\int_{t_0}^{t_f} \left(\sum_{i=1}^{n} (\delta_i - \delta_i^L)^T (\delta_i - \delta_i^U)\right)dt \tag{5.79}$$

where w_1 and w_2 are the corresponding weighting factors, $w \in [0,1]$, for the final downrange longitude and the maximal preference allocation, respectively. Note that the preference for this test case is arbitrarily taken as the middle of the control surface deflection range as represented by the upper and lower limits, δ^U and δ^L, respectively. Table 6.6 compares the cost and the range for various weighting-factors using the 6-DOF Problem SS. Recall, this model consists of a spherical, rotating Earth, and assumes coordinated turns (zero sideslip) and the small flight-angle relation for AoA $(\alpha = \theta - \gamma)$. As previously shown, this flight-angle relation is only valid for the max-downrange case where the bank and roll angles are relatively small.

Table 6.6 Effects of Quadratic Cost - Bootstrapping from 20 to 20 Nodes

w_1	w_2	**Cost 1**	**Cost 2**	**Total Cost,** $J(\cdot)$	**Max DR (nm)**	**CR (nm)**
1.00	0.00	-0.05071	0.00000	-0.05071	174.311	0.0150
0.90	0.10	-0.03657	0.01004	-0.02653	125.717	2.3096
0.75	0.25	-0.03484	0.01830	-0.01654	119.776	2.2988
0.50	0.50	-0.03419	0.03322	-0.00097	117.528	3.3079
0.25	0.75	INFEAS	INFEAS	INFEAS	INFEAS	INFEAS

c. Aerodynamic Model Sensitivities

Numerous aerodynamic functional dependencies present the biggest problem. A common approach for preliminary analysis is to use analytical expression to approximate the aerodynamic data. This involves small perturbation theory to linearize stability and control derivatives about nominal operating points. Depending on the vehicle, there may be certain sensitivities in the aerodynamic data. As a pertinent example for this work, there appears to be an unstable lateral-directional point in the X-33's aerodynamic data that causes a critical flight condition at *M*=3.16 [45]. Although the root cause of this condition was not investigated in the raw data, preliminary results indicate the possibility.

d. Non-Smooth Data

By gradually increasing the nodes, starting with 12, it was determined that the solution failed to converge at 32 nodes and it didn't make a difference if the run was bootstrapped with a converged solution or not. The best solution in terms of computational speed and accuracy was with bootstrapping from 14 to 28 nodes. Using 200 nodes failed to give any results at all after running nearly 8 hours. The most likely cause of this behavior has to do with the non-smooth data and that by increasing the nodes, the gradients are shooting off (diverging) when it tries to capture the discontinuous regions. This reason may also be related to the divergence experienced when trying to propagate the converged solution. An interesting result was found when comparing a low-node solution to a higher bootstrapped solution. The bootstrapped result (not shown) indicated signs of divergence towards the end of the run. If the number of bootstrapped nodes was increased, then this divergence appeared sooner in the data. Again, this confirms the conjecture of problems with handling the aerodynamic data.

As previously described, the main trouble with using table look-up data is that the noise is amplified when computing the Jacobian. One possible way to check this is to use finite differences to compute the Jacobian and analyze the matrix [196]. Significant jumps in the data may predict that too many nodes will not work.

This problem can be resolved by smoothing the original aerodynamic data. Most data from experimental testing contains some form of statistical noise. The objective is to reduce this noise or to at least "discern and highlight patterns concealed in the data" [197]. According to Ref. [197], this is similar to regression analysis in statistics. In flight vehicle applications involving the use of aerodynamic data, the data are often *N*-dimensional, where $N \geq 2$. When $N > 3$, it is nearly impossible to visualize this data. Without a good tool, the process of fitting and/or smoothing large "multi-dimensional data are, in general, theoretically challenging and computationally prohibitive" [197]. As such, only rudimentary aero-date fitting was conducted in this particular research effort as provided in Appendix A. More on this is discussed in the "future work" chapter. One possible improvement without having to smooth the data offline, is to implement an online low pass filter in DIDO.

C. CONCLUDING REMARKS

Although numerical errors associated with the aerodynamic data and possibly effects from over-actuated dynamics hindered a thourough evaluation of the proposed 6-

DOF integrated G&C approach, valuable insight can be gained from the results. As evident from some maximum performance comparisons, there are more substantial effects of using 3-DOF models verses 6-DOF models than originally anticipated. The three main problems encountered in this study were the dependencies on the tabulated aero-database, the implicit control structure, and the high frequency dynamics.

First, as previously explained and demonstrated, the use of tabular data in reentry trajectory optimization is problematic. It is therefore highly recommended that data smoothing and/or more efficient curve-fitting techniques be employed in future work involving 6-DOF trajectory optimization.

Second, since the 6-DOF equations for this problem attempted to use the full, nonlinear functional dependencies in the aerodynamic coefficients, the controls were not explicit in the formulations, rather they were embedded in the functions such as $C = f(\alpha, \beta, M, \delta)$. This limited the development of the necessary conditions of optimality, particularly the Hamiltonian Minimization Condition (HMC). One way that the control variables can appear explicitly in the equations is by routine linearization techniques such that $C = \frac{\partial C}{\partial \delta}\delta$.

Finally, with the sparsity of nodes in the DIDO solutions, it is apparent that some of the high-frequency dynamics are missed. It is probable that high-frequency dynamics are causing the oscillations and divergence found when propagating the controls with an ODE solver using a small step size. In this case, it may be necessary to model an inner-loop control that should attenuate these modes. As such, this falls under the purview of stability augmentation and warrants further investigation.

Although non-favorable trajectories were generated for some 6-DOF cases using exact aero-angle relations, the combination of the HiFi model and an optimal control approach is apparently picking up inherent effects that are otherwise not encountered in 3-DOF models. Bottom line – the X-33 data is very sensitive and can serve as a prime candidate for future research into the coupling between dynamics, control, and optimization.

Since no prior investigation to this extent exists, this research provides valuable information and has highlighted various areas of concern for future efforts.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION

A. SUMMARY OF RESEARCH

This research investigated the use of optimal control techniques for solving high-fidelity, reentry trajectory optimization problems. Recent advances in computational power and numerical methods make this method an attractive alternative for the guidance and control (G&C) systems of future reusable launch vehicles (RLV) that may require accurate, fast, and robust flight path planning, retargeting, and control reconfiguration. RLV G&C also requires the ability to reliably operate in the extremely dynamic and uncertain atmosphere. Due to the nature of unpredictable weather and inherent aerodynamic uncertainty, the G&C method must be robust to exogenous disturbances (e.g.,, wind, density effects, etc.) and internal errors (e.g., noise, numerical error, etc.) As such, the major robustness concern addressed in this research is the method's ability to compensate for large wind gusts and uncertainty.

While traditional approaches to reentry guidance and trajectory design assume that the primary concern is safely reaching the landing site at the correct attitude and energy, this research emphasizes that methods not based on high-fidelity models and optimality can jeopardize the safety of the vehicle. Specifically, high-fidelity, optimal footprint generation is identified as a critical capability required to improve the reliability and effectiveness of RLVs that need to retarget alternate landing sites in the event of a contingency operation. To improve the autonomous, onboard capability of satisfying the final approach requirements, a method was designed that can automatically generate landing constraints for any given runway geometry. From a given runway's location and geometry, a Final Approach Corridor (FAC) is generated and mapped to the optimal trajectory generation program that is assumed to reside in the RLV's flight control system (FCS) onboard computer.

Several initial conditions were used as test cases for investigating the effectiveness and reliability of the open-loop, PS-guidance method combined with the FAC-generation logic. The Space Shuttle Landing Facility at Kennedy Space Center (KSC), Florida was chosen as the primary landing site for the various terminal guidance scenarios. In addition, numerous runways in Florida were selected and tested in a re-targeting feasibility scenario. Although these simulations did not provide quantitative results as would a full Monte-Carlo analysis, they did provide a "proof-of-concept" demonstration of the accuracy, autonomy, and reliability of the proposed method. Another aspect confirmed by these studies, is that the optimal trajectories often portray signs of intelligent behavior. For example, one investigation aimed at yet another reliability test, revealed that the optimal trajectories tend to emulate the energy management designs of the Space Shuttle's Terminal Area Energy Management (TAEM) and Approach-to-Landing phases. This further illustrates the method's viability for onboard, autonomous RLV applications.

In this research, a PS-guidance method was extended to efficiently and effectively provide feedback control for managing large disturbances and uncertainties. In addition to autonomous, open-loop range maximization, path planning, and footprint generation, this method was successfully implemented by successively re-solving the full open-loop optimal control problem in real-time. Validating the supposition that real-time computation of open-loop optimal controls implies closed-loop control, this approach was effective for entry guidance in the presence of hurricane-force wind gusts. The results revealed that this approach is capable of steering a vehicle through wind gusts of approximately 7.7 % the vehicle's total airspeed. In the specific test cases shown, this consists of a sustained wind gust magnitude of approximately 140 mph over a flight-time duration of 20 seconds. Although the direction of the gust was constant and limited to a specific altitude range, it is representative of real wind-shear phenomenon.

A major focus of this research was developing a 6-DOF "integrated" guidance and control architecture as a unified approach to decoupling inner and outer loop interactions that can potentially interfere with performance objectives. This integrated approach involved using a single-loop structure in the control system that determines the optimal control surface deflections (or rates) directly as opposed to the conventional control allocation techniques. Traditional methods depend on the use of separated and often complicated G&C architectures that requires substantial gain tuning/scheduling, linearization, tracking and disturbance rejection techniques, etc. Posing the problem strictly into the optimal control problem formulation and consequently defining a nonlinear programming problem, a PS-based optimization method is used to solve for the control variables while ensuring both feasibility and optimality of the solutions. This approach eliminates many of the laborious techniques needed in the traditional methods.

Primarily as a means of verifying the 6-DOF integrated G&C solutions, a 6-DOF simulation was developed that integrated an outer-loop, 3-DOF optimal trajectory generator and a body-rate command approximation with an inner-loop, reconfigurable tracking controller developed by AFLR. After appropriately tuning the gains and modifying the command generation logic, the inner-loop controller successfully tracked the P,Q,R-body rates converted from an optimal 3-DOF trajectory. Comparing trends of the resulting control histories, appropriate states, as well as calculated forces and moments, helped confirm the feasibility of the 6-DOF optimal trajectory solutions.

Included in this 6-DOF work were examples of problems associated with using the standard 6-DOF equations of motion for trajectory optimization applications. Using a homotopy path as a troubleshooting technique helped demonstrate the sensitivity of the 6-DOF optimization problem to the uncertain aerodynamic database.

In an attempt to capture the inherent high frequency effects due to fast rotational dynamics from the flight vehicle attitude equations, an anti-aliasing Bellman algorithm was applied. This implementation verified that the low-node solution was in fact an alias of a higher accuracy solution. But, more importantly, it revealed that the rates of the control surface deflections exceeded the practical limits of the vehicle; hence, requiring the use of a pseudo-control problem formulation in order to constrain the control rates.

Finally, this research work concluded by comparing the 3-DOF and 6-DOF performance, specifically the maximum range trajectories and the approximate maneuverability envelopes based on a fixed initial condition corresponding to a flight over the Gulf of Mexico. This comparison re-affirmed the claim that basing footprints on lower fidelity models is dangerous if implemented in a real vehicle or even if used for offline trajectory design and/or trajectory reshaping. Although the differences were not as great as those found from others in previous studies, they were enough to make landing infeasible. The average maximum differences between the 3-DOF (without trim effects) model and the 6-DOF model were approximately 4.24 % (~8 nm) in downrange and 13.3 % (~14 nm) in crossrange.

B. ORIGINAL CONTRIBUTIONS

This research provided a new, unified approach for autonomous G&C of the highly constrained and nonlinear RLV reentry problem. By applying optimal control theory, the reentry problem was solved by the DIDO/SNOPT sparse optimization software package. Although the reentry problem itself is fundamentally the same, the approach is unique in that it avoids the common trajectory segmentation, trimmed flight, and reference trajectory tracking used throughout the field. Rather, it solves the complete trajectory as one problem using the same formulation and algorithm. With its real-time capability, new trajectories are generated throughout the flight eliminating the reliance on tracking a reference trajectory or scheduling/storing/executing pre-programmed maneuvers.

By the nature of the optimal control formulation, there are no transfer functions to be designed or gains to be tuned. Likewise, only the physically realistic constraints of the problem are modeled and as such do not put unnecessary limitations on the true physics of the system. For example, it is common in reentry G&C to use trimmed flight constraints and zero sideslip constraints.

Furthermore, it is important to stress that this approach is suitable for any flight vehicle requiring the use of the full 6-DOF equations of motion. Though the focus on this research was the RLV reentry application, most of the formulation, code, FAC-target logic, and PS algorithm are equally valid for any flight vehicle including the automatic trajectory generation for commercial or general aviation aircraft. In this respect, the approach offers a vehicle-independent solution to a plethora of flight problems.

A second unique contribution of this research is the implementation of the PS-feedback method to the RLV reentry problem. Due to the anti-aliasing effect, it was determined that a relatively low degree of discretization (i.e., number of LGL-node points) is sufficient for closed-loop optimal guidance. Thus, it is apparent that this technique is viable for use in optimal guidance algorithms that require corrective maneuvers from the perturbed trajectory. Perhaps more importantly, this feedback implementation contributed to the verification and validation of new theoretical developments in the field of pseudospectral-based optimal control. The PS-feedback guidance verified the requirements of a sampling frequency for generating Carathéodory-π solutions and illustrated the detrimental effects of using a less sophisticated feedback control such as the more common sample-and-hold techniques (e.g., ZOH).

To date, there have been only two attempts to solve the highly representative, 6-DOF flight equations of motion using direct trajectory optimization methods. The first, to the best of the author's knowledge, employed a similar PS-method to compute optimal combat flight maneuvers; however, this work implemented the standard EoM [195]. Concurrently with the research presented herein, a two-timescale collocation method was employed for a class of RLV trajectory optimization problems [78],[79]. Although these efforts both solved 6-DOF problems, this work is unique because it not only extends the method to a feedback implementation, but that it also exemplifies the approach as a viable option for a new autonomous G&C architecture that can provide speed, guaranteed convergence, and robustness.

The importance of this demonstrably successful application to the reentry problem is three fold. First, it validates that PS methods can easily generate optimal reference trajectories for highly dynamical nonlinear aerospace problems. Secondly, it shows promise for solving a rich variety of dynamical systems by breaking from the traditional concept of feedback control and paving the way towards a unified approach employing real-time optimal control. Finally, with such powerful methods readily available and the expanding computational speed of computers, why settle for less than optimal? As implied by the various examples in this dissertation, optimality not only provides savings in terms of time, cost, control, fuel, etc, but also improves safety, reliability, robustness, and autonomy as typically required of intelligent systems.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. OPEN ISSUES AND FUTURE WORK

A. MODELING AND FIDELITY

In terms of flight dynamics, a large portion of this work focused on deciphering various formulations that existed in textbooks, literature, and flight mechanics software. With this, most of the time was spent performing rudimentary analysis of the flight equations of motion and coordinate systems. Emphasis was placed on understanding the various assumptions used in the derivations such that some of these assumptions could be removed when modeling the full 6-DOF equations. For example, numerous contradictions, mostly as a result of different notation and assumptions, inspired further investigation into the flight angle relationships as summarized in Chapter III. The intent was to use a more correct, complete equation formulation; however, a full derivation and methodical examination of the equations was not a purview of this work. With this said, a more thorough investigation into the flight equations along with numerical testing is required to select the most appropriated equations suitable for a high fidelity optimization problem. Similarly, the models used for the earth (shape and gravity) and its atmosphere were not modeled to their highest fidelity. Although the higher-order terms are often negligible, it is possible that combined effects of various simplifying assumptions can have unexpected results. Essentially, there is a sensitivity factor involved just as there is for the aerodynamic data that has inherently large uncertainties.

B. DATA SMOOTHING AND MULTI-DIMENSIONAL DATA-FITTING

The 6-DOF aerodynamic data used in this work is approximated via AFRL's linear interpolation of the tabulated database. As emphasized by Betts, the use of linear interpolation is "by far the single most catastrophic impediment to an efficient solution of the trajectory optimization problem" [64]. As such, this dissertation work initially involved the use of numerous curve-fitting and surface-fitting techniques and tools to better approximate the aerodynamic coefficients (see Appendix A). Unfortunately, time precluded the validation and implementation of accurate curve-fits, and instead the original linear interpolation was employed.

As previously mentioned, experimental flight testing (e.g., wind tunnel testing) often contains numerous errors and uncertainties that manifest as non-smooth data. In some cases, these uncertainties are not trivial. For example, an X-33 aerodynamic uncertainty study modeled uncertainties as high as 80% [173]. This degree of uncertainty along with the inherent discontinuous nature of tabulated data helps explain some of the computational problems encountered in the 6-DOF trajectory optimization work when using the X-33 tabulated aerodynamic database. Overall, the use of non-smooth data in optimization codes can present various problems. Some of these problems, as mentioned throughout this dissertation and as pointed out by Shaffer's work [46], include singularities in the gradients, long calculation times, and "near optimal" verses "locally optimal" solutions. Because of this, accurate and robust multi-dimensional data fitting (e.g., 2D-curve or 3D-surface-fitting) techniques are needed. As recommended by Betts, a viable method to smooth tabular data is a cubic B-spline or a minimum curvature spline to more accurately represent "qualitative aspects" of the data [115]. Note that MATLAB offers a spline toolbox that has a function for B-splines called "bspline" and a function for 3D data smoothing called "smooth3." Another promising technique for efficient smoothing of tabular data involves the use of Linear-Quartic Chamfer Splines [198]. Note that Moerder gratefully provided the "LQspline" code, but unfortunately time precluded its use in this research work. In any case, it is a worthwhile endeavor to use refined data and/or accurate and computationally tractable approximations for future high fidelity flight trajectory optimization.

C. HIGH FREQUENCY DYNAMICS

As discussed in Chapter VI results, one of the problems encountered from using non-smooth data was failure of DIDO to give reliable results when using a high number of nodes (n > 30). It is common practice with discrete optimization software to increase the number of nodes (i.e., mesh refinement) in order to improve the accuracy of the results. Obtaining this accuracy usually comes at the penalty of increased computational time since the scale of the problem increases as a function of the number of nodes. For example, the scale of the problem is represented by the number of discrete optimization variables given by $N_n(N_x + N_u)+1$, such that the 6-DOF model in this work for a 20-

node case has 401 optimization variables. For a problem that has intrinsically different dynamic timescales, such as most flight vehicles with their slow translational dynamics and fast rotational dynamics, increasing the number of nodes should capture the high-frequency effects. Since this problem was unable to successfully bootstrap to higher nodes, the anti-aliasing technique was instead implemented to recover the high-frequencies in the control histories. If data smoothing is not performed, it may be prudent to compare the anti-aliasing technique with other so called time-scale separation techniques. One possible technique worth pursuing is a two-timescale discretization scheme that Desai developed for use with a collocation method [78]. His approach provides an avenue for reducing problem size while still maintaining a high accuracy solution. Valuable insight may be gained by implementing a similar approach using the pseudospectral method employed in this research. Regardless, additional research is needed to confirm the conjecture that non-smooth data causes the “high-node” problem.

D. IMPROVEMENTS FOR COMPUTATIONAL SPEED

Fast computational speed may be a critical requirement for most aerospace applications using optimal control techniques such as the PS method. Since the runtime for the 6-DOF reentry problem is excessive for on-board, real-time applications, it is worth exploring run-time reduction methods. As such, it is compulsory to make a few suggestions on areas that could help enhance performance. The two elements hindering speed are the computer platform (hardware and software) and the un-optimized algorithms used in the DIDO and MATLAB software.

1. Computer Platform

Obviously the computer’s CPU directly influences the speed of numerical calculations. Most of this work was performed on a Dell Optiplex GX260 with a Pentium 4 CPU, 3.06 GHz, and 1.05 GB RAM. Although this CPU and memory is sufficient for computationally intensive algorithms, the Windows and server environment (2000, SP 4) required approximately 50 running processes that consumed nearly half of the memory resources and 100 % of the CPU performance while running MATLAB 7.0.1. Moon concluded that transferring the DIDO code from the Windows/Server environment into a dedicated LINUX machine and from the MATLAB software to C-code, can increase the speed by at least a factor of 100 [199]. As far as the DIDO code itself, there are numerous opportunities.

2. Algorithms in DIDO Code

Future improvements to the DIDO algorithm also have the potential of significantly reducing computational time. Three general areas worth exploring are: 1.) algorithmic choices based on convexity/non-convexity, 2.) the use of analytical Jacobians, and 3.) warm start procedures.

Additional features that would improve DIDO, but not necessarily effect runtime, are the following:

1. Automatic scaling
2. Automatic mesh refinement (i.e., bootstrapping)
3. Automatic verification of satisfying necessary optimality conditions
4. Automatic propagation for feasibility analysis
5. Automatic Bellman optimality tests

Also, DIDO code can be made to accommodate more flexible input files. For example, when using the path function file, it is only setup to pass in certain data (e.g., "primal" array), which consequently may require the path file to reproduce computations that were performed elsewhere. Such was the case in using the aerodynamic database. The tables had to be interrogated not only once in the dynamics file, but also in the path file. This unnecessary duplication of function evaluations can impact time. Also, DIDO requires the first and last node guesses to coincide with the specified knot locations. This is a limitation that may have a quick fix.

E. LIST OF SPECIFIC TASKS FOR FUTURE WORK

The following list contains some of the specific tasks, in random order, that can be performed as a direct continuation of the work presented in this dissertation.

- Explore the use of various curve-fitting techniques
- Smooth the aerodynamic data

- Develop accurate approximations to the aerodynamics using “explicit” control variables in equations of motion instead of “implicit” functional dependencies
- Apply the approach to other RLV models (recommend X-38 since actual flight data exists)
- Perform more analysis with nonzero sideslip angle (use exact flight equations)
- Conduct various failure scenarios and compare to previous work
- Improve scaling (consider Lagrange multipliers)
- Use quaternion attitude representation instead of Euler angle states in equations of motion
- Improve wind gust model and integrate into vehicle model dynamics
- Perform more rigorous Monte-Carlo analysis for the 3-DOF feedback-implementation (e.g., sustained winds, random winds, turbulence, etc.)
- Integrate path constraints into the cost function to improve robustness of feedback-implementation
- Perform stability and sensitivity analysis
- Implement feedback into the 6-DOF problem
- Implement terminal guidance logic (i.e., FAC) for 6-DOF problem

Although this is not an exhaustive list, it reflects some of the more important issues identified in this dissertation.

F. EXTENDED APPLICATIONS

With minimum modifications, this research work can be extended to solving various other flight-based applications requiring trajectory optimization. The following sections summarize some potential extended applications that could be pursued with additional follow-on efforts.

1. RLV Landing Footprint Generation and Retargeting

As briefly demonstrated in this dissertation (see Chapter IV), real-time generation of maximum performance footprints is well within reach. With the capability to quickly generate landing footprints, pareto-optimal fronts, a vehicle in distress can determine which landing sites are feasible. With the addition of a voting algorithm or by simply comparing performance costs, the preferable site can be targeted. This autonomous capability presents the best option (i.e., optimal) in the advent that the originally intended landing site is not reachable.

2. Planetary Reentry, Descent, and Landing

Designing vehicles and their flight control methods to land on Mars is an active area of research, especially after the recent U.S. presidential initiative promoting a return to the moon and eventually trips to Mars. Assuming the same equations of motion are applicable on other planets, the approach developed in this dissertation is readily extendable to other planetary missions, including Mars entry, descent, and pin-point landing. As a unified, vehicle-independent method, the presented PS-based reentry G&C should only require minimal modifications for Mars missions. In a "plug-and-play' fashion, this may consist of only swapping the planet model, atmosphere model, aerodynamic model, and vehicle parameters.

3. Launch Vehicle Ascent and Reusable Fly-Back Boosters

Obviously this optimal G&C approach is not limited to reentry, but can also be applied to ascent applications including all launch vehicles and ballistic missiles. For example, Rea has demonstrated the applicability of PS methods for launch vehicle ascent [123]. Going even further, the open-loop method has successfully solved a multi-stage launch vehicle problem as well as the hybrid optimal control of a "two-agent" vehicle for mission planning applications [200]. Adding the PS-feedback capability demonstrated in this dissertation to be a viable approach for flight vehicles, these applications can now be advanced to the next level by providing real-time, closed-loop endo-atmospheric guidance and/or control.

Similar to the reentry problem, the "fly-back" guidance for a reusable booster designed to land itself at a designated site can also use this approach. As such, the booster would be able to recompute an optimal trajectory in the case of some off-nominal flight condition. How about migrating this to a more "every-day" application, such as commercial airliners?

4. Contingency Path Planning and Landings for Commercial Airliners

An interesting and possibly near-term application of the proposed approach involves providing commercial airliners a real-time, onboard contingency path planning option. Consider a commercial airliner, or any fixed-wing airplane for that matter, flying into the vicinity of its final destination when it is determined that severe thunder storms preclude it from landing. Typically, the pilot needs to coordinate with air traffic control (ATC) and determine a feasible alternate landing site. In accordance with recommended Federal Aviation Administration (FAA) practices, any good pilot will already have an alternate landing site planned, but it is not uncommon to divert to unplanned airports, especially with unpredicted weather, system failures, etc. With an onboard version of this optimal trajectory generation either embedded in the aircraft's flight computer or on a separate handheld computer, the pilot could use the tool to generate a set of feasible landing sites in fractions of a second as well as an optimal candidate flight path that avoids the thunder storms. The pilot could then choose to manually track a set of optimal waypoints provided by the computer or have the autopilot do so. This can even go as far as optimally scheduling and de-conflicting en-route flights.

5. Path Planning for UAVs and MAVs

On a smaller scale, the method can be extended for use in UAV and MAV path planning. With the military focusing on more unmanned, intelligent combat systems, the development of new vehicle G&C methods is essential. For example, the Army's Future Combat Systems (FCS) architecture is highly dependent on autonomous and semi-autonomous vehicle capabilities [201] that includes onboard mission planning and execution. Lewis demonstrated that the PS-based optimal control approach used in this work can also successfully solve various obstacle avoidance problems using ground vehicle models and a simplified UAV model [178]. Integrating the full 6-DOF flight equations, along with some other ideas presented in this dissertation, into his work can provide a very powerful capability.

Overall, the possibilities are endless!

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A - AERODYANMIC COEFFICIENT APPROXIMATIONS

A. CHARACTERIZATION OF X-33 AERO DATA (AVERAGE CURVES)

In order to get a very crude approximation of the functional dependency effects on the individual aerodynamic coefficieints, various average curves were generated. The following sets of two-dimensional (2D) curves are separated into contributions from the control surface deflections and contributions from the base effects only, i.e., the nominal "clean-body" configuration with no control deflections. Although angle-of-attack (α), sideslip angle (β), and Mach number (*M)* are variables that change throughout the flight regime, in order to simplify the multi-dimensional data tables, these variables were fixed to arbitrary constants based on the expected range of operations. Note that at higher speeds, the vehicle is likely at higher α and lower β.

1. Contributions from Control Surface Deflections

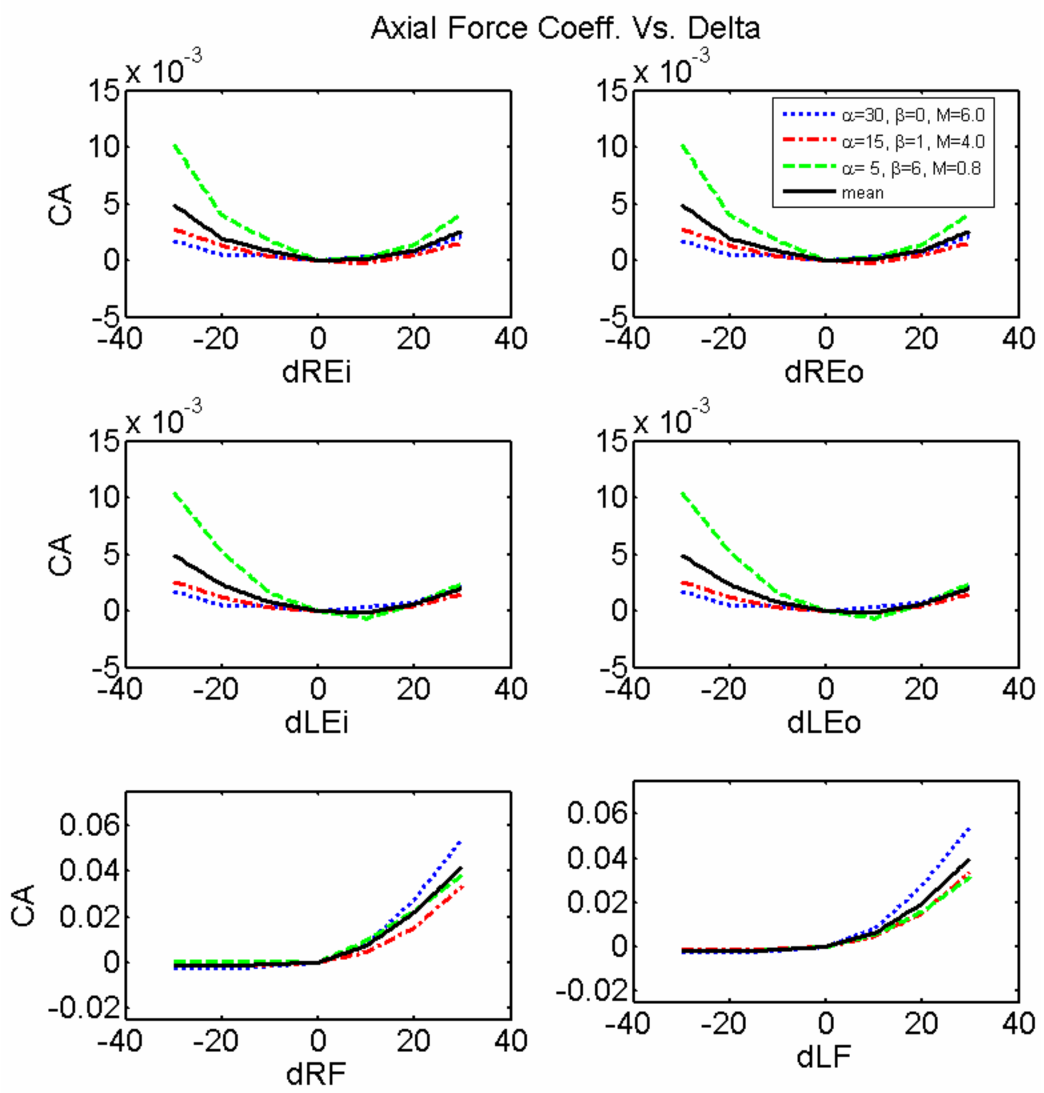


Figure A.1 Axial Force Coefficient vs. Control Surface Deflections.

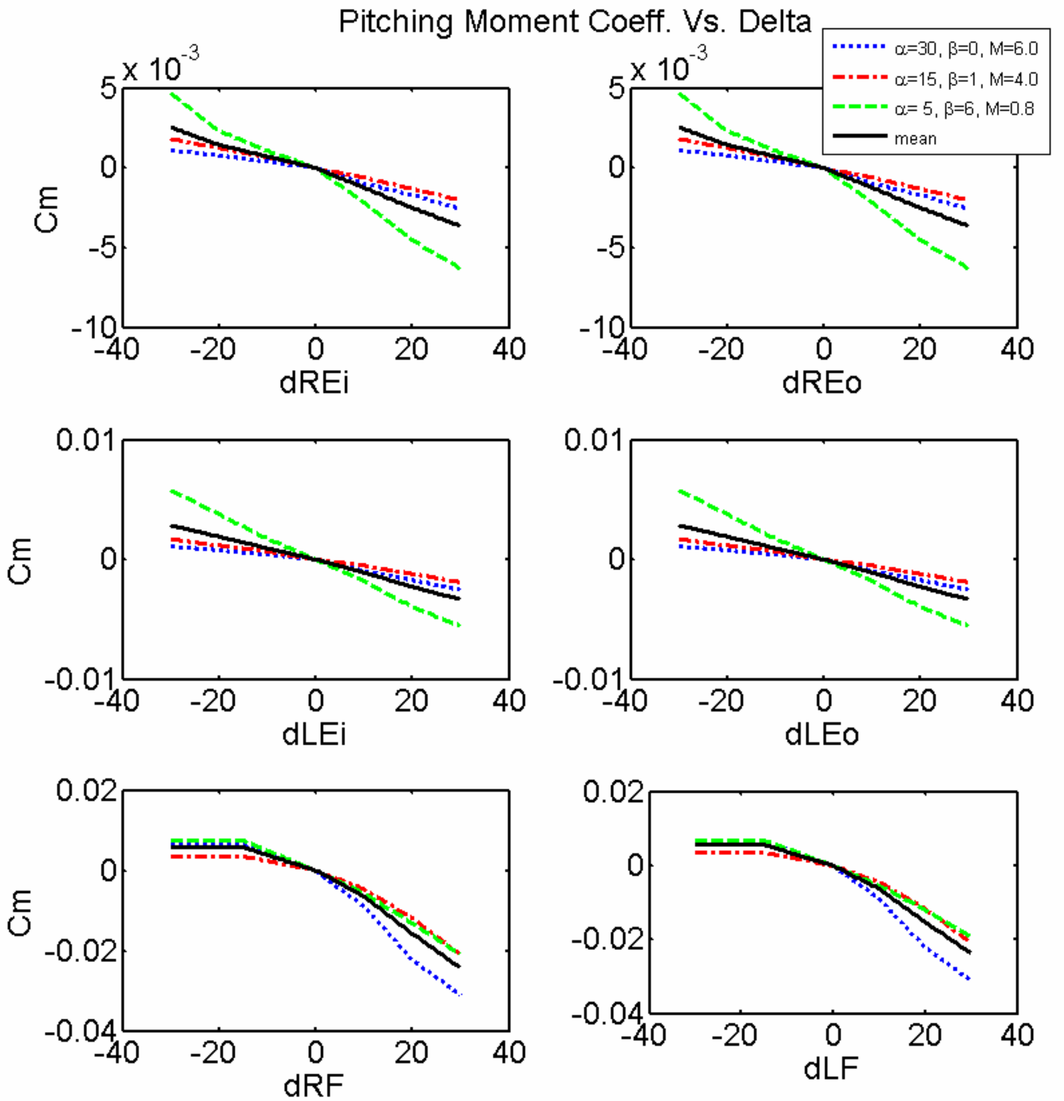


Figure A.2 Pitching Moment Coefficient vs. Control Surface Deflections.

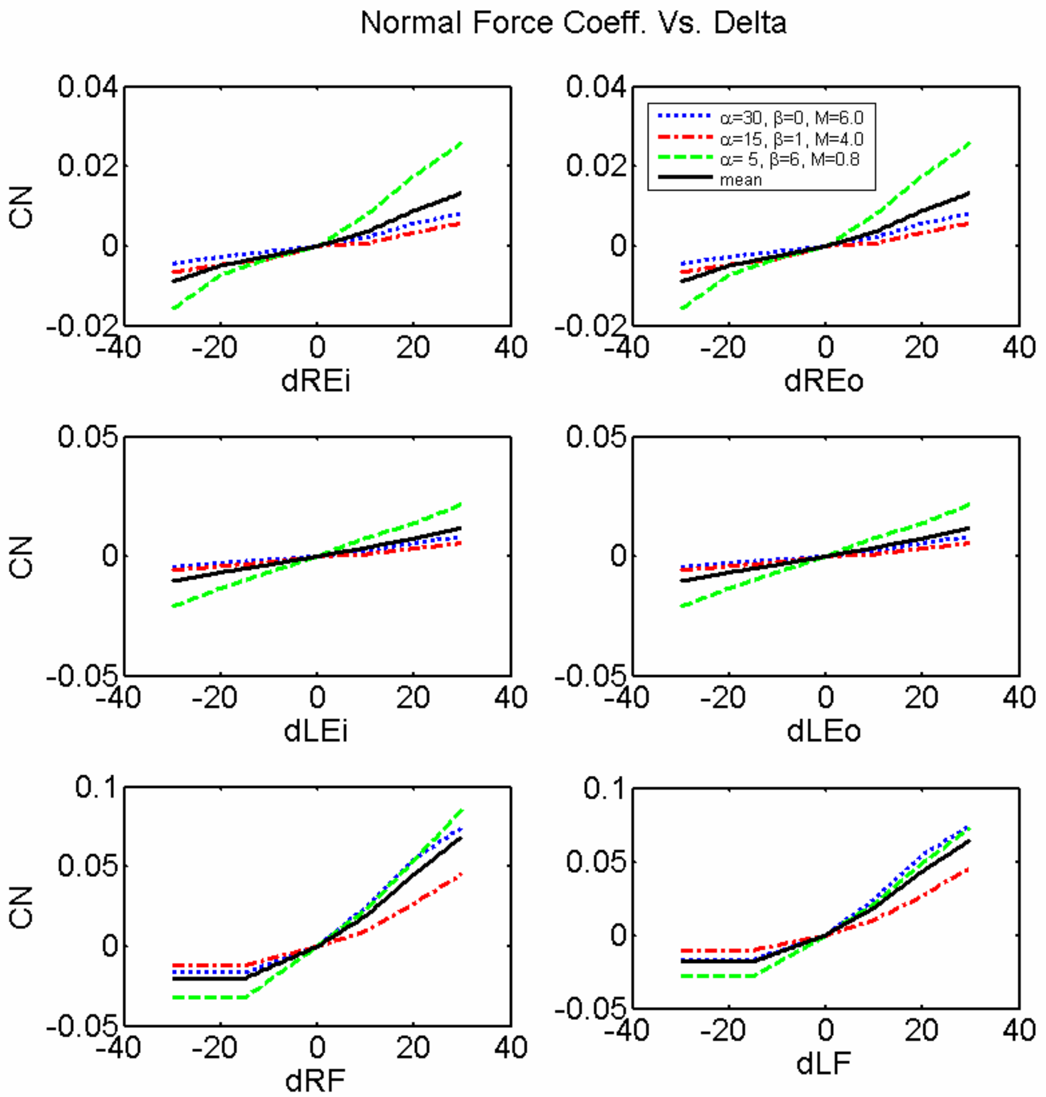


Figure A.3 Normal Force Coefficient vs. Control Surface Deflections.

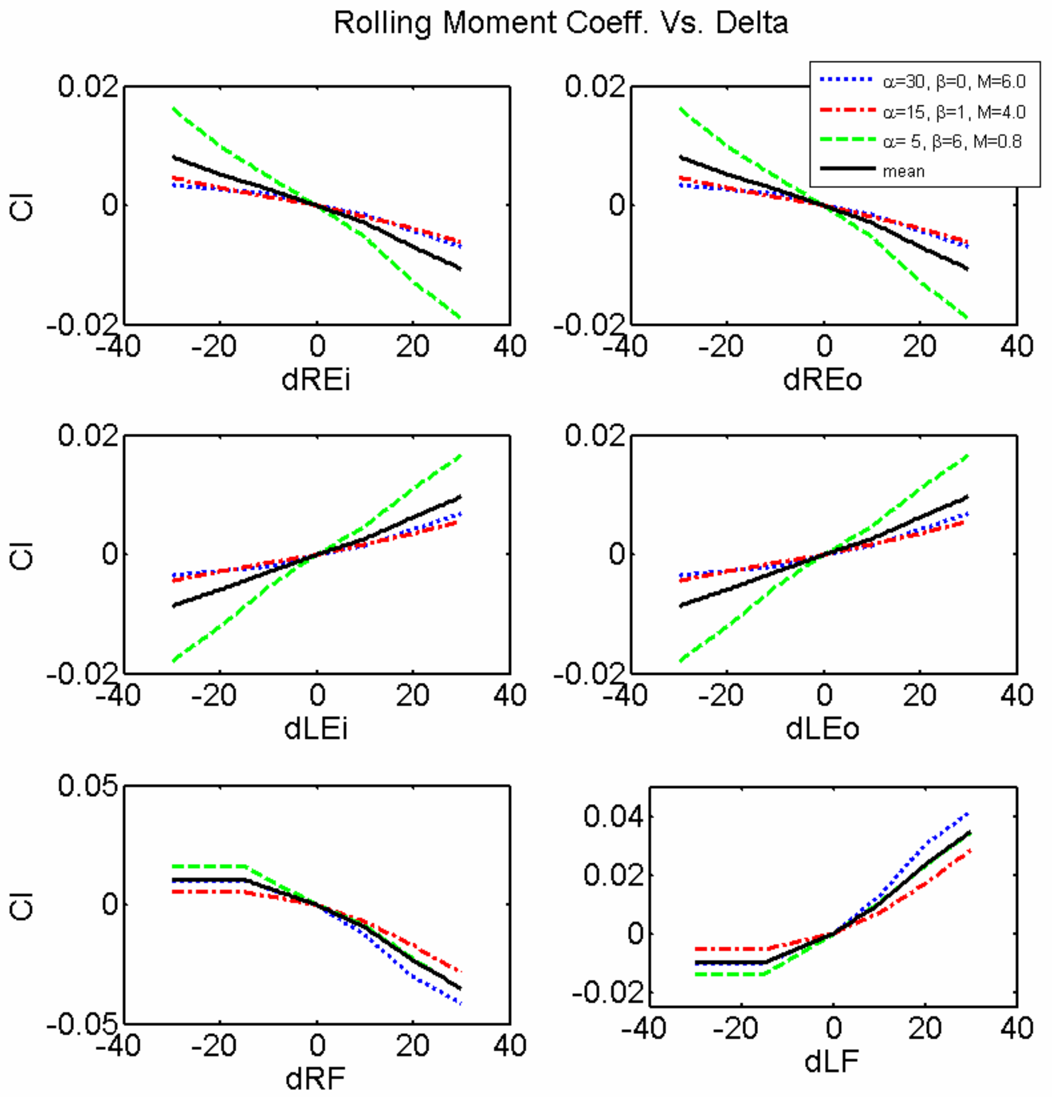


Figure A.4 Rolling Moment Coefficient vs. Control Surface Deflections.

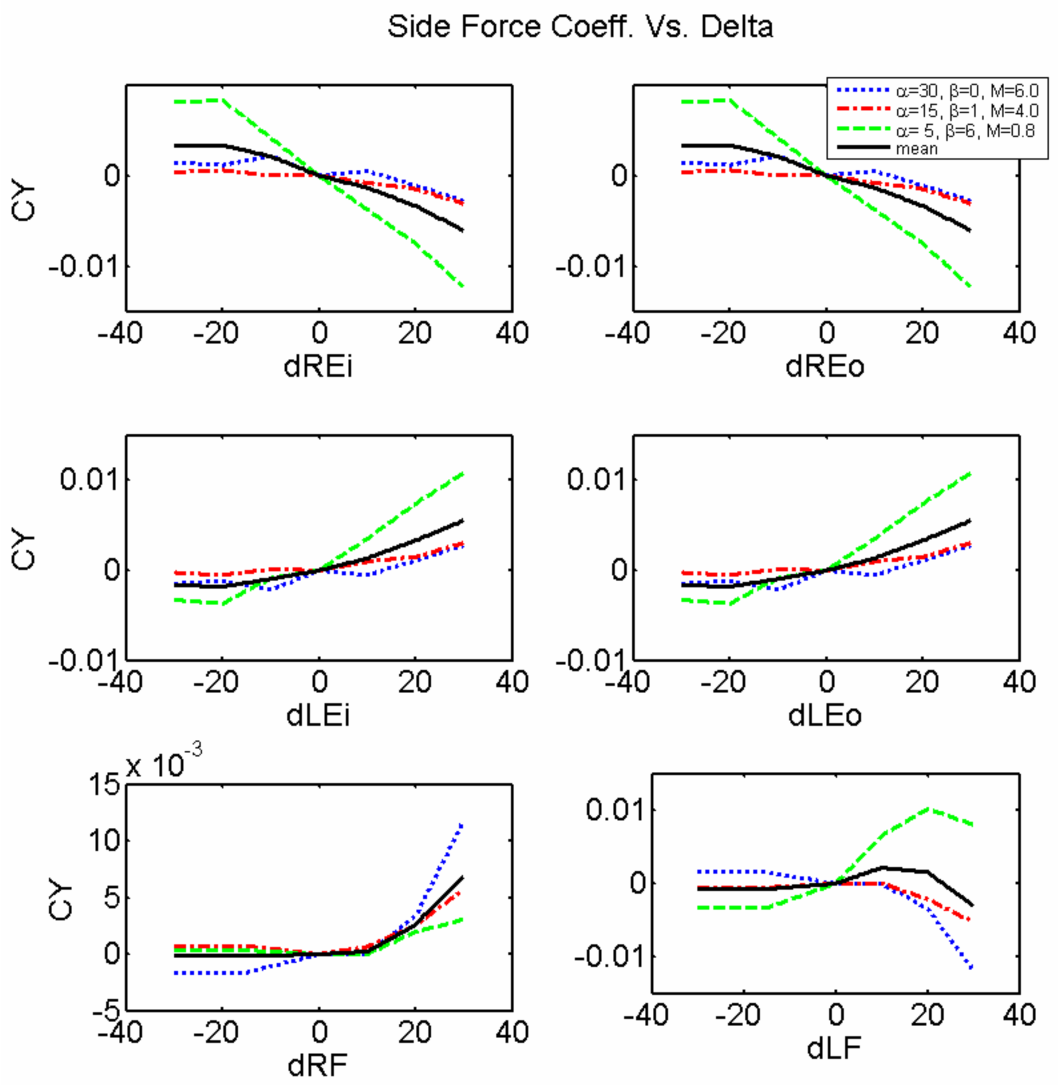


Figure A.5 Side Force Coefficient vs. Control Surface Deflections.

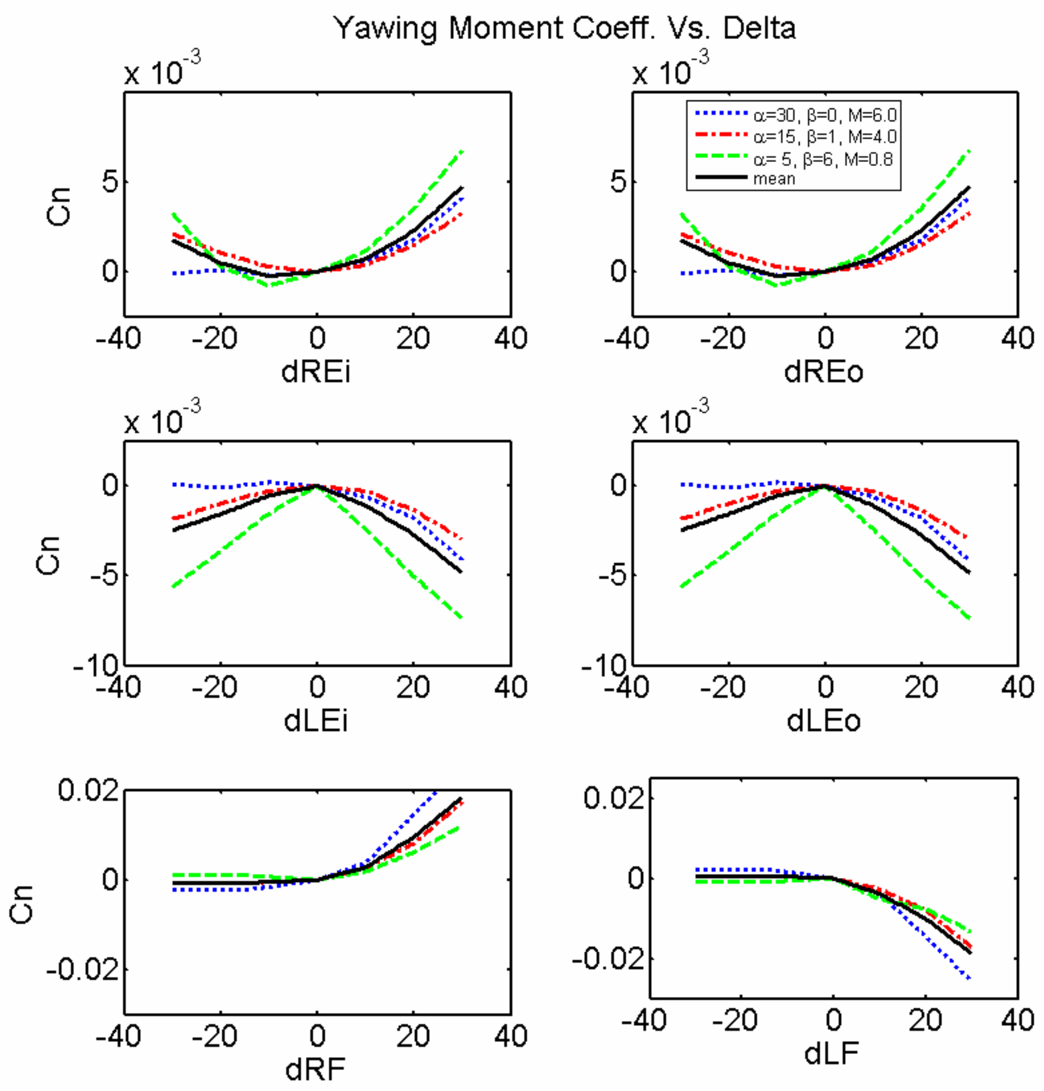


Figure A.6 Yawing Moment Coefficient vs. Control Surface Deflections.

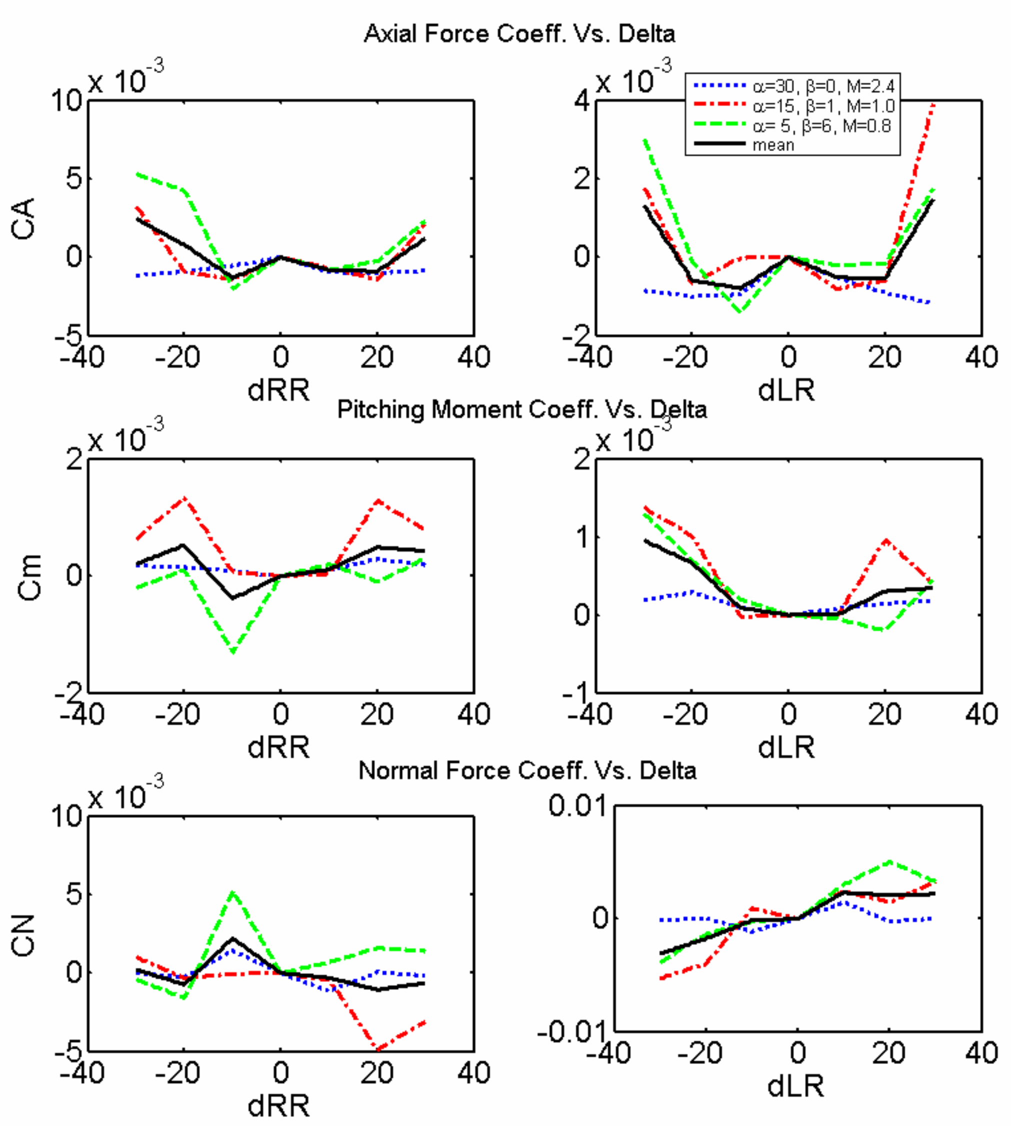


Figure A.7 Longitudinal Coefficients vs. Rudder Deflections.

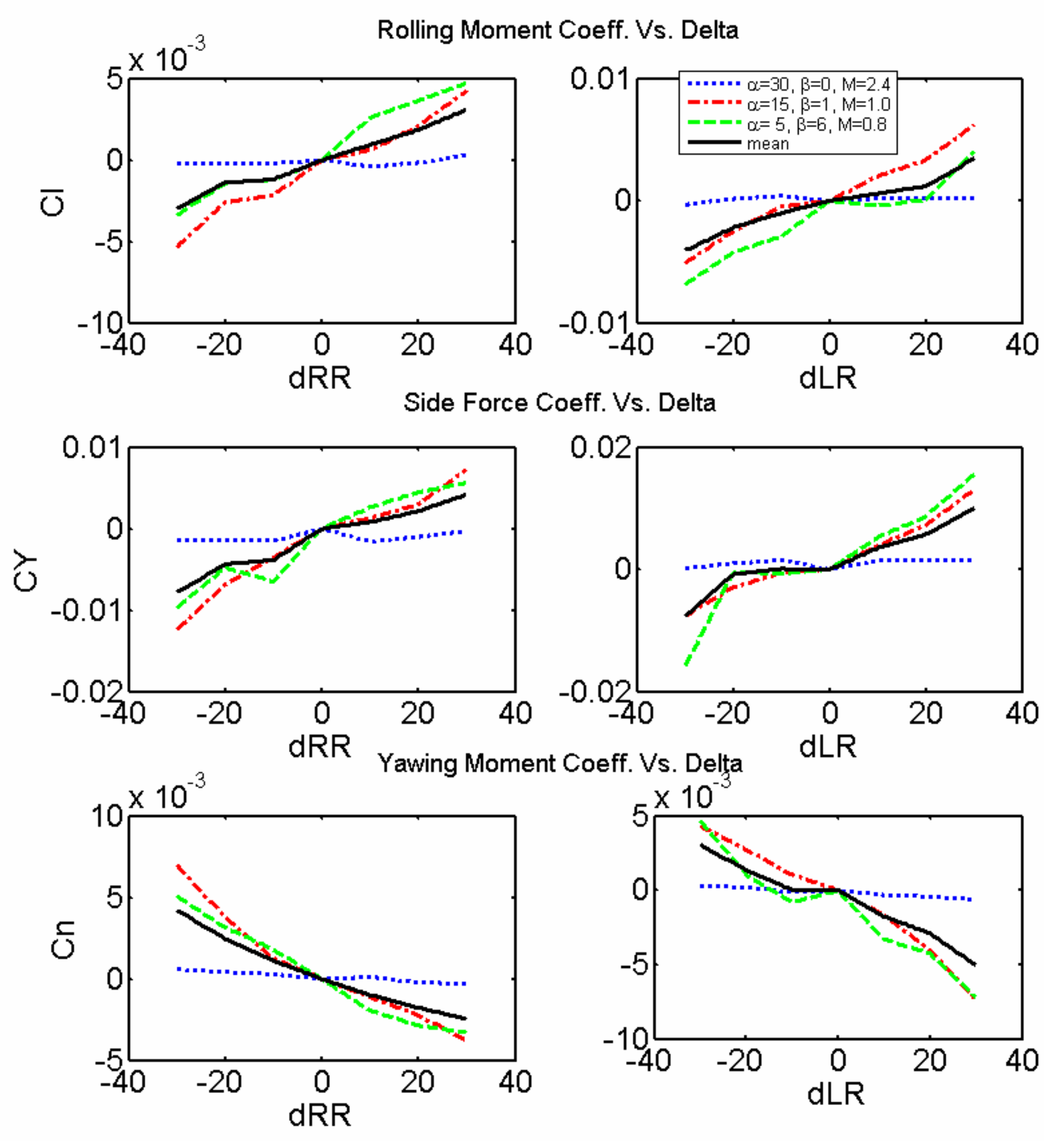


Figure A.8 Lateral-Directional Coefficients vs. Rudder Deflections.

Table A.1 Polynomial Curve Fits for Aero Force Coefficients

		Aerodynamic Coefficient: y=f(x)	
		CA	Norm Error
Control Surface Positions (x)	REi	y = 4.0025e-006*x^{2} - 3.2939e-005*x + 5.6058e-005	0.0012522
	REo	y = 4.0025e-006*x^{2} - 3.2939e-005*x + 5.6058e-005	0.0012522
	LEi	y = 3.8948e-006*x^{2} - 4.593e-005*x + 1.9524e-006	0.0005670
	LEo	y = 3.8948e-006*x^{2} - 4.593e-005*x + 1.9524e-006	0.0005670
	RF	y = 2.2536e-005*x^{2} + 0.00062504*x + 0.0011774	0.0167790
	LF	y = 2.1133e-005*x^{2} + 0.00059166*x + 0.00057217	0.0172500
	RR	y = 2.6593e-006*x^{2} - 2.5452e-005*x - 0.00085771	0.0045559
	LR	y = 1.6368e-006*x^{2} + 3.3976e-006*x - 0.00065776	0.0039314

		Aerodynamic Coefficient: y=f(x)	
		CY	Norm Error
Control Surface Positions (x)	REi	y = - 1.6727e-006*x^{2} - 0.00016304*x + 0.0003978	0.0023242
	REo	y = - 1.6727e-006*x^{2} - 0.00016304*x + 0.0003978	0.0023242
	LEi	y = 2.1443e-006*x^{2} + 0.00012305*x + 9.6197e-006	0.0006723
	LEo	y = 2.1443e-006*x^{2} + 0.00012305*x + 9.6197e-006	0.0006723
	RF	y = 3.8511e-006*x^{2} + 8.6248e-005*x - 0.00016625	0.0052527
	LF	y = - 2.3242e-006*x^{2} + 2.2627e-005*x + 0.00078711	0.0102700
	RR	y = - 1.0672e-006*x^{2} + 0.00018801*x - 0.00084201	0.0044227
	LR	y = 8.382e-007*x^{2} + 0.0002251*x + 0.001378	0.0115220

		Aerodynamic Coefficient: y=f(x)	
		CN	Norm Error
Control Surface Positions (x)	REi	y = 2.7395e-006*x^{2} + 0.00035235*x + 0.00037582	0.0049520
	REo	y = 2.7395e-006*x^{2} + 0.00035235*x + 0.00037582	0.0049520
	LEi	y = 9.2712e-007*x^{2} + 0.00036395*x + 1.8599e-006	0.0013861
	LEo	y = 9.2712e-007*x^{2} + 0.00036395*x + 1.8599e-006	0.0013861
	RF	y = 2.8028e-005*x^{2} + 0.001574*x + 0.00041808	0.0150860
	LF	y = 2.6754e-005*x^{2} + 0.0014795*x + 0.00094174	0.0158860
	RR	y = - 1.4511e-006*x^{2} - 2.3643e-005*x + 0.0004284	0.0068906
	LR	y = - 1.2584e-006*x^{2} + 9.5217e-005*x + 0.00070911	0.0032958

Table A.2 Polynomial Curve Fits for Aero Moment Coefficients

		Aerodynamic Coefficient: y=f(x)	
		Cl	**Norm Error**
Control Surface Positions (x)	**REi**	y = - 1.5896e-006*x^{2} - 0.00030589*x - 4.0881e-005	0.0019617
	REo	y = - 1.5896e-006*x^{2} - 0.00030589*x - 4.0881e-005	0.0019617
	LEi	y = 7.7506e-007*x^{2} + 0.000302*x - 0.00010903	0.0013473
	LEo	y = 7.7506e-007*x^{2} + 0.000302*x - 0.00010903	0.0013473
	RF	y = - 1.4337e-005*x^{2} - 0.00081152*x - 0.00012868	0.0150860
	LF	y = 1.4325e-005*x^{2} + 0.0007999*x + 0.00048483	0.0089242
	RR	y = 2.8219e-007*x^{2} + 9.3139e-005*x - 3.6642e-005	0.0015436
	LR	y = - 3.8597e-007*x^{2} + 0.00010282*x - 0.00016134	0.0032802

		Aerodynamic Coefficient: y=f(x)	
		Cm	**Norm Error**
Control Surface Positions (x)	**REi**	y = - 5.8299e-007*x^{2} - 0.00010072*x - 0.00017672	0.0009949
	REo	y = - 5.8299e-007*x^{2} - 0.00010072*x - 0.00017672	0.0009949
	LEi	y = - 2.6121e-007*x^{2} - 0.00010383*x - 4.2181e-005	0.0002748
	LEo	y = - 2.6121e-007*x^{2} - 0.00010383*x - 4.2181e-005	0.0002748
	RF	y = - 1.0538e-005*x^{2} - 0.00052302*x - 0.00036032	0.0040355
	LF	y = - 1.0295e-005*x^{2} - 0.00050748*x - 0.00036886	0.0037943
	RR	y = 6.7472e-007*x^{2} + 4.0132e-006*x - 2.7661e-005	0.0018112
	LR	y = 8.4018e-007*x^{2} - 9.0623e-006*x + 3.279e-005	0.0007375

		Aerodynamic Coefficient: y=f(x)	
		Cn	**Norm Error**
Control Surface Positions (x)	**REi**	y = 3.6656e-006*x^{2} + 4.7162e-005*x - 2.3198e-005	0.0005359
	REo	y = 3.6656e-006*x^{2} + 4.7162e-005*x - 2.3198e-005	0.0005359
	LEi	y = - 3.9587e-006*x^{2} - 3.376e-005*x - 0.00040244	0.0019552
	LEo	y = - 3.9587e-006*x^{2} - 3.376e-005*x - 0.00040244	0.0019552
	RF	y = 9.9348e-006*x^{2} + 0.00027436*x + 0.000368	0.0078677
	LF	y = - 9.7579e-006*x^{2} - 0.00028749*x - 0.00070484	0.0061287
	RR	y = 9.8117e-007*x^{2} - 0.00010827*x - 1.0871e-005	0.0004818
	LR	y = - 6.2845e-007*x^{2} - 0.00011779*x - 0.00052043	0.0032878

2. Contributions from Base Effects Only – Nominal "Clean-Body" Configuration

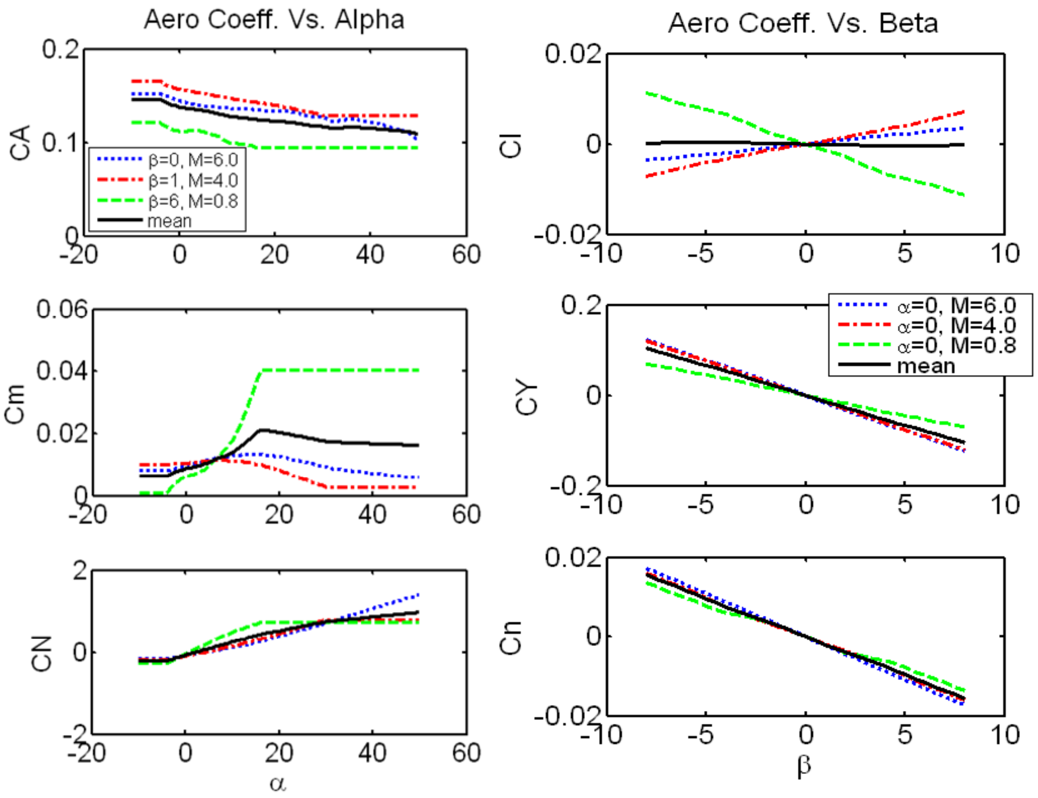


Figure A.9 Base Coefficients vs. Alpha and Beta.

Table A.3 Curve Fits for Base Aero Coefficients

			Aerodynamic Coefficient: y=f(x)	Norm Error
Functional Dependencies	**α**	**CA**	y = 8.6459e-006*x^{2} - 0.00096188*x + 0.13843	0.0122700
		Cm	y = - 1.0107e-005*x^{2} + 0.00059259*x + 0.010142	0.0177650
		CN	y = - 0.0002107*x^{2} + 0.030784*x - 0.01851	0.3942300
	β	**Cl**	y = - 5.4905e-005*x - 1.7694e-020	0.0007385
		CY	y = - 0.013165*x - 3.6237e-018	0.0034410
		Cn	y = - 0.0019305*x - 6.0395e-019	0.0007935

3. Contributions from Damping Derivatives

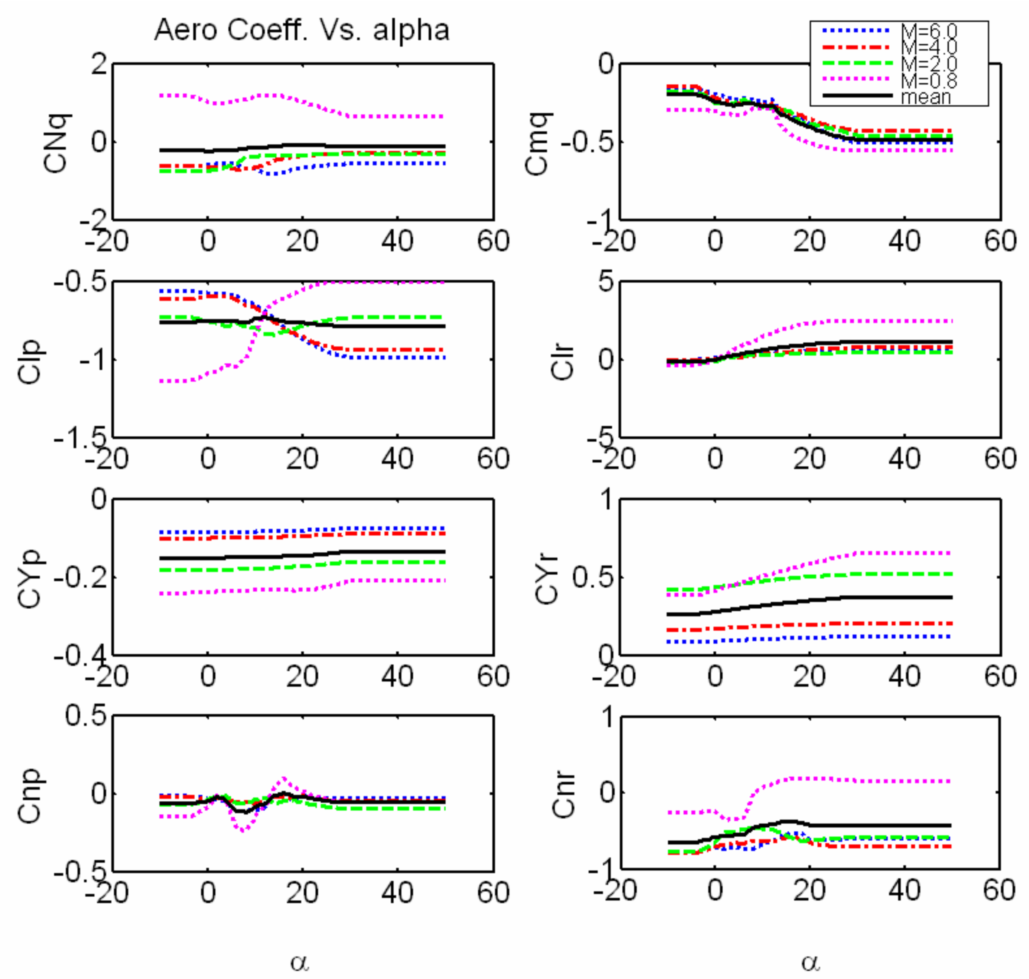


Figure A.10 Damping Derivative Coefficients vs. Alpha.

Table A.4 Curve Fits for Damping Derivative Aero Coefficients

			Damping Derivatives: y=f(x)	Norm Error
Functional Dependencies	**α**	**CNq**	y = - 8.4979e-005*x^{2} + 0.0051351*x - 0.19193	0.266430
		Cmq	y = 7.2108e-005*x^{2} - 0.0091332*x - 0.23968	0.318500
		Clp	y = - 1.2446e-005*x^{2} - 0.00031204*x - 0.75563	0.107440
		Clr	y = - 0.00064472*x^{2} + 0.050787*x + 0.15032	0.789490
		CYp	y = 9.3034e-008*x^{2} + 0.00041547*x - 0.15098	0.022320
		CYr	y = - 4.3145e-005*x^{2} + 0.0039895*x + 0.28568	0.063453
		Cnp	y = - 2.2604e-005*x^{2} + 0.0010361*x - 0.058175	0.245790
		Cnr	y = - 0.00017095*x^{2} + 0.010418*x - 0.56246	0.337600

B. A LEAST "SUM-OF-SQUARES" ERROR APPROXIMATION

Another method used to approximate the aerodynamic coefficients was an algorithm adapted from Keshmiri's "FITTER" code that generates polynomial fits to flight vehicle data based on minimizing a sum-of-square error, a traditional approach [202]-[204]. The routine essentially finds the best fitting equation (out of ten predictions) for any input aerodynamic matrix in the form of $[\alpha, M, \delta]$. By measuring the deviation of the fitted values from the actual data, the algorithm picks the best possible fit based on the performance of the sum-of-square error. Due to the drastically different behavior of the data for hypersonic verses supersonic or subsonic speeds, it is numerically better to fir the data separately. Although it is probably more effective to separate into the three categories mentioned, for this testing the data was only separated into two sets: hypersonic $(M \geq 5)$ and any speed less $(M < 5)$. Since the ……… , this method proved very time consuming and after running some tests using actual flight data inputs did not seem consisten with the accuracy measurements used to select the best-fit equations. Some fits tended to have substantially higher errors than others. As such, an alternate method was sought to complement the coeeficients with higher errors. For example, an online curve/surface-fitting program called *ZunZun*© was used as well [205].

C. SURFACE-FITTING MULTI-DIMENSIONAL TABULATED DATA

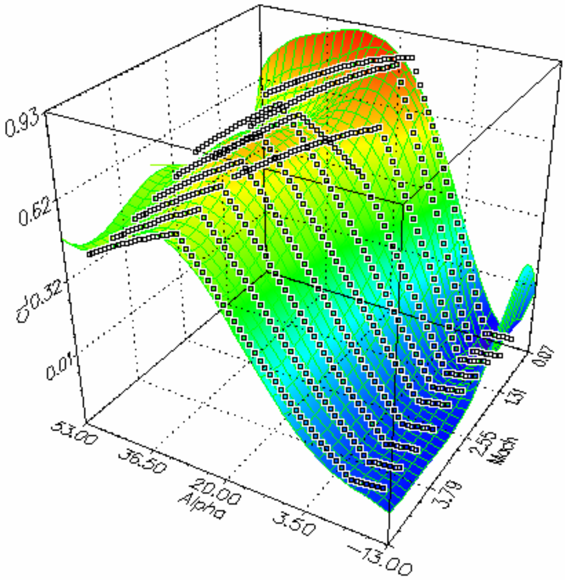
To help compensate for some of the visual inadequacies associated with the previous two aerodynamic approximation methods, two more tools were explored. First, an on-line tool called *ZunZun* [205] was used to fit three-dimensional (3D) tabulated aero data. This very simple yet powerful tool has the capability to plot, analyze, and fit data. The user only needs to enter the 2D or 3D data in the appropriate format and can then select what actions to take including options from a statistical toolbox. Including the same fitting-criterion as the FITTER algorithm previously mentioned, this tool offers the benfit of visually verifiying the results. For example, the following results were generated by inputting the aerodynamic table data for coefficieint of lift (C_L) for a Mach range of 0 to 4.8. Using the lowest sum-of-squared error, the results were as follows:

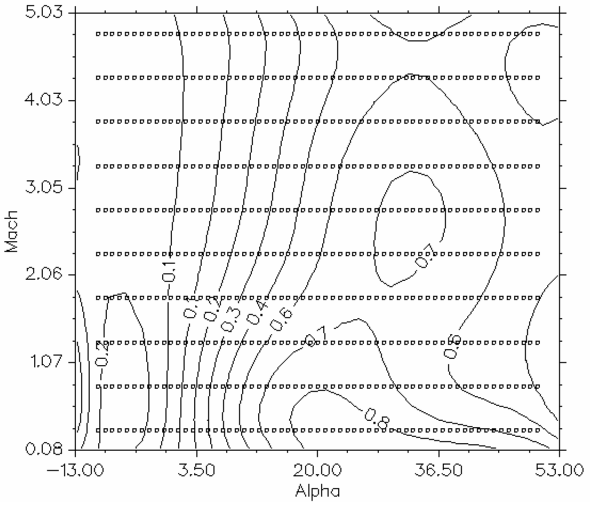
Data Statistics

	X	Y	Z
Minimum:	-1.000000E+01	3.000000E-01	-2.369700E-01
Maximum:	5.000000E+01	4.800000E+00	8.711100E-01
Mean:	2.000000E+01	2.550000E+00	3.641397E-01
Median:	2.000000E+01	2.550000E+00	4.672600E-01
Sample Variance:	3.100000E+02	2.062500E+00	9.852651E-02
Sample Std Dev:	1.760682E+01	1.436141E+00	3.138893E-01
Pop. Variance:	3.105090E+02	2.065887E+00	9.868830E-02
Pop. Std Dev:	1.762127E+01	1.437319E+00	3.141469E-01
Variation:	8.803408E-01	5.631924E-01	8.620024E-01
Skew:	0.000000E+00	5.745097E-14	-4.767033E-01
Kurtosis:	-1.200645E+00	-1.224242E+00	-1.084190E+00

5th Order Polynominal:

$z = a + bx^0y^1 + cx^0y^2 + dx^0y^3 + ex^0y^4 + fx^0y^5 + gx^1y^0 + hx^1y^1 + ix^1y^2 + jx^1y^3 + kx^1y^4 + lx^1y^5 + mx^2y^0 + nx^2y^1 + ox^2y^2 + px^2y^3 + qx^2y^4 + rx^2y^5 + sx^3y^0 + tx^3y^1 + ux^3y^2 + vx^3y^3 + wx^3y^4 + xx^3y^5 + yx^4y^0 + zx^4y^1 + Ax^4y^2 + Bx^4y^3 + Cx^4y^4 + Dx^4y^5 + Ex^5y^0 + Fx^5y^1 + Gx^5y^2 + Hx^5y^3 + Ix^5y^4 + Jx^5y^5$





Error Statistics

	Absolute Error	Relative Error
Minimum:	-1.160733E-01	-5.790913E+00
Maximum:	1.056640E-01	3.734828E+00
Mean:	-1.967930E-07	-1.765218E-02
Median:	2.337568E-04	3.669358E-03
Sample Variance:	1.207538E-03	1.321617E-01
Sample Std Dev:	3.474965E-02	3.635405E-01
Pop. Variance:	1.209521E-03	1.323787E-01
Pop. Std Dev:	3.477817E-02	3.638389E-01
Variation:	-1.765797E+05	-2.059465E+01
Skew:	2.480964E-01	-5.566786E+00
Kurtosis:	1.757675E+00	1.280627E+02

Similar to *ZunZun,* another commercially available 3D surface-fitting tool is *TableCurve3D©* [206]. It provides slightly more advanced capabilities including the ability to automatically generate code in various programming languages. Figure A.11 is an example of a TableCurve3D surface-fit for the X-33 drag coefficient data.

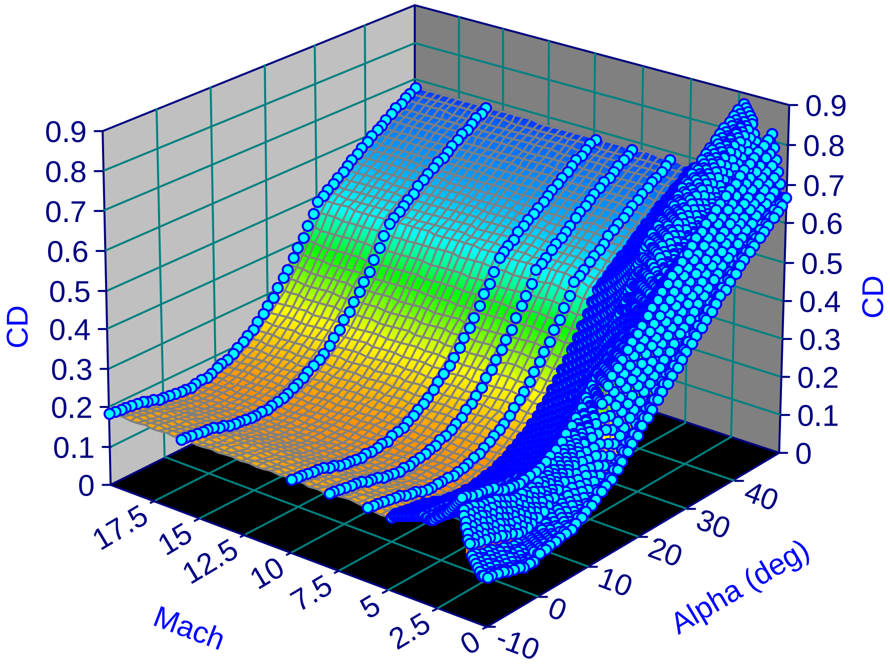
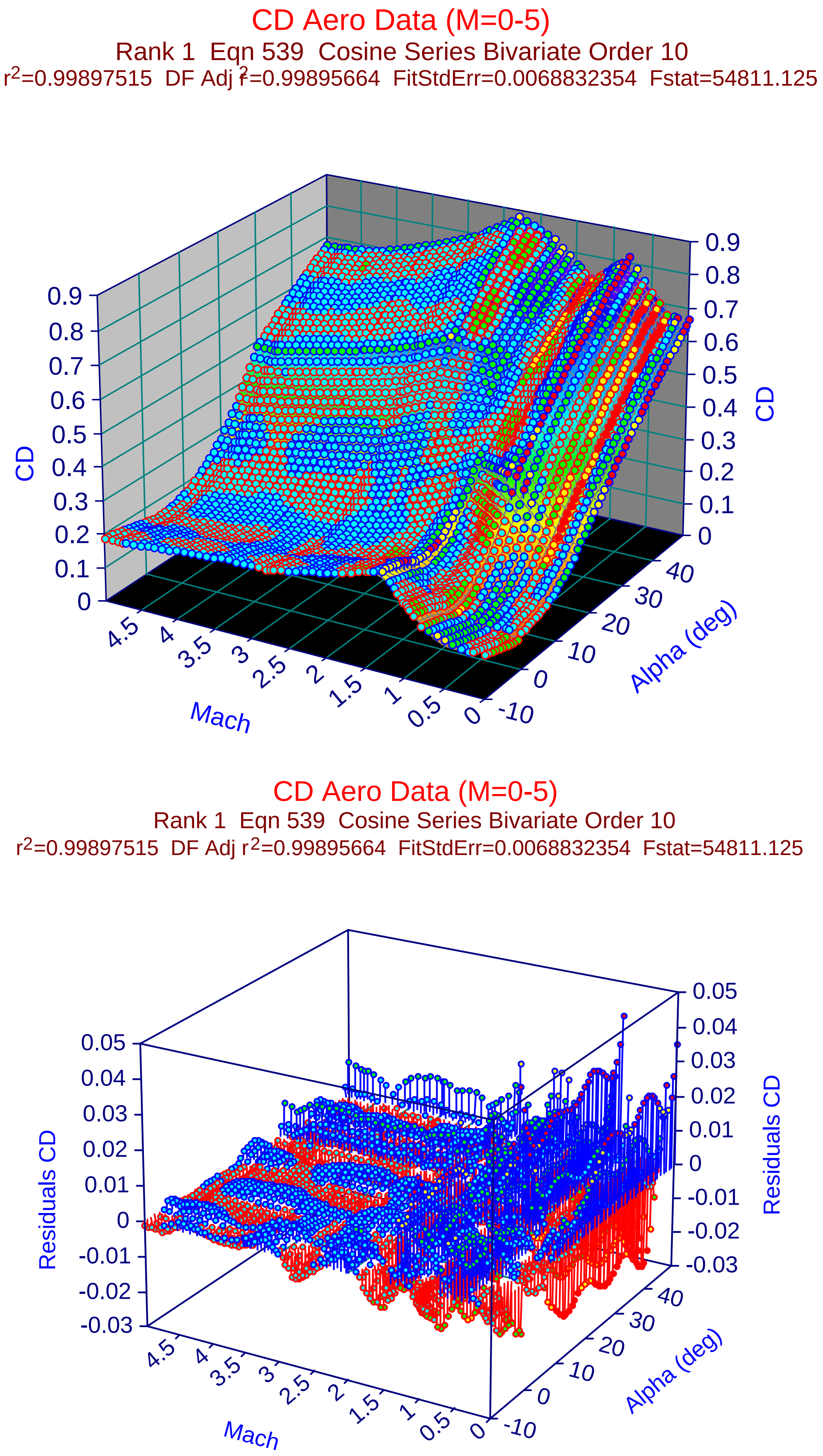


Figure A.11 Example of TableCurve3D Surface-Fit to Drag Coefficient.

Also available is various statistical tools including visual representation of the error residuals as shown by Figure A.12.



CD Aero Data (M=0-5)

Rank 1 Eqn 539 Cosine Series Bivariate Order 10

r^2=0.99897515 DF Adj r^2=0.99895664 FitStdErr=0.0068832354 Fstat=54811.125

CD Aero Data (M=0-5)

Rank 1 Eqn 539 Cosine Series Bivariate Order 10

r^2=0.99897515 DF Adj r^2=0.99895664 FitStdErr=0.0068832354 Fstat=54811.125

Figure A.12 Example of TableCurve3D Error Residuals.

```
Rank 1  Eqn 539  Cosine Series Bivariate Order 10

r2 Coef Det     DF Adj r2       Fit Std Err     F-value
0.9989751507    0.9989566395    0.0068832354    54811.124645

 Parm   Value          Std Error      t-value        95.00% Confidence Limits     P>|t|
  a     0.405276446    0.000115175    3518.780405    0.405050632    0.40550226    0.00000
  b     -0.28437301    0.000161324    -1762.74958    -0.28468931    -0.28405672   0.00000
  c     0.021442695    0.000162891    131.6380453    0.021123328    0.021762062   0.00000
  d     0.053621811    0.00016111     332.8274476    0.053305937    0.053937685   0.00000
  e     -0.01219486    0.000227824    -53.5276053    -0.01264154    -0.01174819   0.00000
  f     -0.05113431    0.000164881    -310.129234    -0.05145758    -0.05081104   0.00000
  g     0.019450103    0.000160699    121.0346038    0.019135035    0.019765171   0.00000
  h     -0.02222281    0.000226617    -98.0633405    -0.02266711    -0.0217785    0.00000
                                              Etc…

X at Fn Zmin    Y at Fn Zmin    Fn Zmin
-1.62842093     0.4151426864    0.0976251652
X at Fn Zmax    Y at Fn Zmax    Fn Zmax
50              2.2082302069    0.8595816875

Procedure
GaussElim
r2 Coef Det     DF Adj r2       Fit Std Err
0.9989751507    0.9989566395    0.0068832354
Source    Sum of Squares    DF      Mean Square        F Statistic       P>F
Regr      168.79801         65      2.5968924          54811.1           0.00000
Error     0.17316999        3655    4.737893e-05
Total     168.97118         3720

Description: CD Aero Data (M=0-5)

X Variable: Alpha (deg)
   Xmin:     -10               Xmax:    50              Xrange:   60
  Xmean:     20                Xstd:    17.60918321

Y Variable: Mach
   Ymin:     0                 Ymax:    5               Yrange:   5
  Ymean:     2.4124590164      Ystd:    1.4490503008

Z Variable: CD
   Zmin:     0.095326542       Zmax:    0.87741056      Zrange:   0.782084018
  Zmean:     0.41003383        Zstd:    0.2131252214

Date            Time            File Source
Oct 26, 2005    2:02:01 PM      h:\nps files\h drive files\phd research
```

APPENDIX B - MODEL FIDELITY CHARTS

The following charts complement the discussions throughout this dissertation involving model and method fidelity/complexity and a homotopy path to modeling. Figure D.1 shows a conceptual evolution of the GNC architecture based on the model and the methods. Moving from left to right in the architecture, models are improving fidelity and methods are becoming more "general."

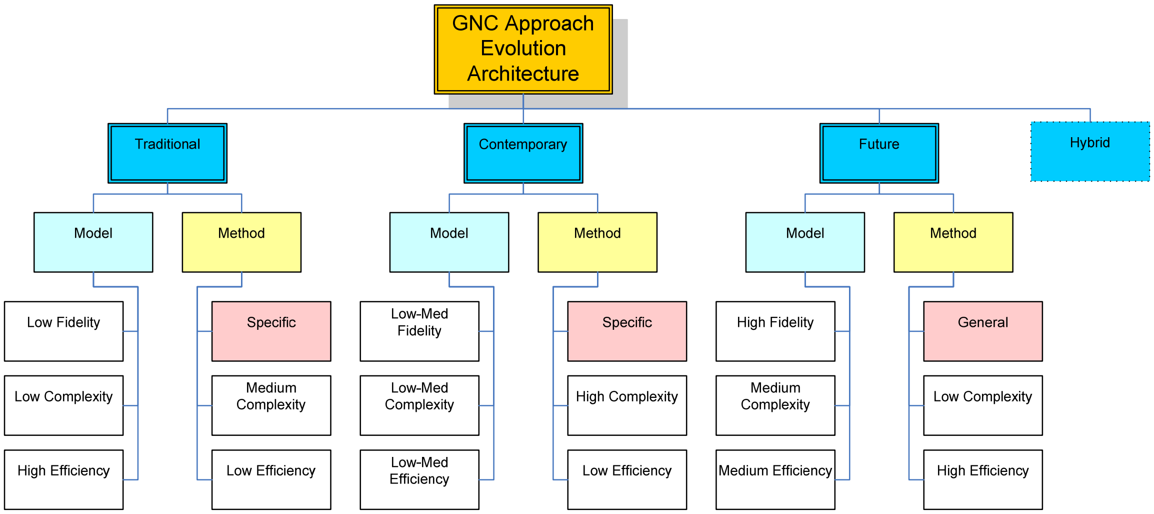


Figure D.1 GNC Approach Evolution Architecture

Table D.1 shows a candidate process for sequentially increasing model fidelity from a 2-DOF to a 6-DOF launch vehicle model.

Table D.1 Homotopy Path to High-Fidelity 6-DOF Model

Model 1	Model 2	Model 3	Model 4
point mass / (2-DOF)	point mass / (3-DOF)	point mass (4-DOF)	rigid body (6-DOF)
flat earth	flat earth	flat earth	flat earth
constant gravity	constant gravity	constant gravity	variable gravity
constant density	exponential density	exponential density	density table (atmos66)???
no heat constraint	no heat constraint	approx. heat constraint	heat constraint
no control limits	control limits	control limits	control limits
no g-load constraint (normal)	no g-load constraint (normal)	g-load constraint (normal)	g-load constraint (normal)
no q-load constraint (axial)	no q-load constraint (axial)	q-load constraint (axial)	q-load constraint (axial)
no side-force constraint	no side-force constraint	no side-force constraint	side-force constraint
Cl & Cd (alpha) closed-form	Cl & Cd (alpha) closed-form	Cl & Cd (alpha, M) -surface fit	Cl & Cd (alpha, M) -table???
pure longitudinal motion	longitudinal & lateral motion	longitudinal & lateral motion	long/latl/direct/rotational motion
no disturbances	no disturbances	no disturbances	no disturbances
no uncertainties	no uncertainties	no uncertainties	no uncertainties
open loop	open loop	open loop	open loop

Model 5	Model 6	Model 7
rigid body (6-DOF)	rigid body (6-DOF)	rigid body (6-DOF)
spherical/non-rotating	spherical/rotating	oblate spheroid/rotating
variable gravity	variable gravity	variable gravity
density table (atmos66)???	other???	other???
heat constraint	heat constraint	heat constraint
control limits	control limits	control limits
g-load constraint (normal)	g-load constraint (normal)	g-load constraint (normal)
q-load constraint (axial)	q-load constraint (axial)	q-load constraint (axial)
side-force constraint	side-force constraint	side-force constraint
Cl & Cd (alpha, M) -table??	other???	other???
long/latl/direct/rotational motion	long/latl/direct/rotational motion	long/latl/direct/rotational motion
no disturbances	disturbances (wind model)	disturbances (wind model)
no uncertainties	uncertainties???	model uncertainties & sensor noise
closed loop???	closed loop???	closed loop

🟨 = changed from previous model

LIST OF REFERENCES

[1] Friedrich, H.R., and Dore, F.J., "The Dynamic Motion of a Missile Descending Through the Atmosphere," *Journal of Aeronautical Sciences*, Vol. 22, pp. 628-632, 1955.

[2] Allen, H.J., and Eggers, A.J., Jr., "A Study of the Motion and Aerodynamic Heating of Missiles Entering the Earth's Atmosphere at High Supersonic Speeds," NACA TN 4047, 1957.

[3] Boutelle, J., Kau, S.P., and Marino, CJ., "ICBM Reentry Vehicle Navigation System Development at Honeywell," *Position Location and Navigation Symposium*, IEEE, pp. 294-302, Palm Springs, CA, April 1998.

[4] U.S. Army, http://www.redstone.army.mil/history/systems/redstone/welcome.html, (accessed August 2006).

[5] London, J. R., *Leo on the Cheap: Methods for Achieving Drastic Reductions in Space Launch Costs*, Air University Press, Maxwell AFB, AL, October 1994.

[6] NASA, "Project Apollo: A Retrospective Analysis," http://www.hq.nasa.gov/office/pao/History/Apollomon/Apollo.html, (accessed October 2006), October 2004.

[7] NASA, *National Space Transportation System – Overview*, NASA History Division [online], http://history.nasa.gov/shuttleoverview1988/overview88.htm, (accessed October 2006), September 1988.

[8] Hannigan, R. J., *Spaceflight in the Era of Aero-Space Planes*, Kreiger Publishing Co., Malabar, FL, 1994.

[9] Ansari X-Prize, http://www.xprize.org/xprizes/ansari_x_prize.html, (accessed October 2006).

[10] NASA, http://www.nasa.gov/mission_pages/constellation/orion/index.html, NASA, (accessed October 2006).

[11] Wade, M., "CEV," *Astronautix* [online], http://www.astronautix.com/craftfam/cev.htm, (accessed October 2006).

[12] Oduntan, G., "The Never Ending Dispute: Legal Theories on the Spatial Demarcation Boundary Plane between Airspace and Outer Space," *Hertfordshire Law Journal*, Vol. 1(2), pp. 64-84, 2003.

[13] NASA, *National Space Transportation System Reference*: Systems and Facilities, Vol. 1, June 1988.

[14] Harpold, J.C., and Graves, C.A., Jr., "Shuttle Entry Guidance," *The Journal of Astronautical Sciences*, Vol. XXVII, No. 3, pp. 239-268, September 1979.

[15] Wingrove, R.C., "Survey of Atmosphere Re-Entry Guidance and Control Methods," *AIAA Journal*, Vol. 1, No. 9, pp. 2019-2029, September 1963.

[16] NASA, *NSTS 1988 News Reference Manual*, NASA, June 1988.

[17] Walyus, K.D., and Dalton, C., "Approach and Landing Simulator for Space Shuttle Orbiter Touchdown Conditions," *Journal of Spacecraft and Rockets*, Vol. 28, No. 4, pp. 478-485, 1991.

[18] Tsikalas, G.M., "Space Shuttle Autoland Design," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. A82-40432, August 1982.

[19] Kafer, G.C., "Space Shuttle Entry/Landing Flight Control Design Description," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. A82-40430, August 1982.

[20] Barton, G.H., and Tragesser, S.G., *Autolanding Trajectory Design for the X-34*, based on paper presented at AIAA Atmospheric Flight Mechanics Conference, August 1999.

[21] Ha, C., and Kim, J-K, "Automatic Landing Under Wind Shear Turbulence in Adaptive Gain Scheduled PID Control Optimized in Genetic Algorithm," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2005-6346, August 2005.

[22] Looye, G. and Joos, H-D., "Design of Autoland Controller Functions with Multiobjective Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, pp. 475-484, April 2006.

[23] Kluever, C.A., "Unpowered Approach and Landing Guidance Using Trajectory Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, pp. 967-974, August 2004.

[24] Miyazawa, Y., Motoda, T., Izumi, T., and Hata, T., "Longitudinal Flight Control for Space Vehicle's Automatic Landing," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 98-4428, August 1998.

[25] Motoda, T., Miyazawa Y., Ishikawa, K., and Izumi, T., "ALFLEX Flight Simulation Analysis and Flight Testing," *Proceedings of the AIAA 36th Aerospace Sciences Meeting*, AIAA Paper No. A98-16195, January 1998.

[26] Stevens B.L., and Lewis, F.L., *Aircraft Control and Simulation,* 2nd Ed., John Wiley & Sons, Inc., New Jersey, 2003.

[27] Roenneke, A.J., and Markl, A., "Re-Entry Control to a Drag-vs-Energy Profile," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 5, pp. 916-920, October 1994.

[28] Lu, P. "Entry Guidance and Trajectory Control for Reusable Launch Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, pp. 143-149, February 1997.

[29] Lu, P. and Hanson, J.M., "Entry Guidance and for the X-33 Vehicle," *Journal of Spacecraft and Rockets*, Vol. 35, No. 3, pp. 342-349, June 1998.

[30] Lu, P., "Regulation About Time-Varying Trajectories: Precision Entry Guidance Illustrated," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 6, pp. 784-790, December 1999.

[31] Shen, Z., and Lu. P., "Onboard Generation of Three-Dimensional Constrained Entry Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 1, pp. 111-121, February 2003.

[32] Ito, D., Ward, D., and Valasek, J., "Robust Dynamic Inversion Controller Design and Analysis for the X-38," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2001-4380, August 2001.

[33] Caplin, J., "Descent-Phase Control of a Reusable Launch Vehicle," *Proc. of the IEEE 21st Digital Avionics Systems Conference*, Vol. 2., pp. 9D1-11, 2002.

[34] Costa, R.R., Chu, Q.P., and Mulder, J.A., "Reentry Flight Controller Design Using Nonlinear Dynamic Inversion," *Journal of Spacecraft and Rockets*, Vol. 40, No. 1, pp. 64-71, February 2003.

[35] Johnson, E.N., Calise, A.J., and El-Shirbiny, A., "Feedback Linearization with Neural Network Augmentation Applied to X-33 Attitude Control," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2000-4157, August 2000.

[36] Johnson, E.N., and Calise, A.J., "A Six Degree-of-Freedom Adaptive Flight Control Architecture for Trajectory Following," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2002-4776, August 2002.

[37] Mease, K.D., and Kremer, J-P., "Shuttle Entry Guidance Revisited Using Nonlinear Geometric Methods," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, pp. 1350-1356, December 1994.

[38] Hanson, J.M., and Jones, R.E., "Test Results for Entry Guidance Methods for Space Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, pp. 960-966, December 2004.

[39] Oppenheimer, M.W., and Doman, D.B., "Reconfigurable Control Design for the X-40A with In-Flight Simulation Results," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2004-5017, August 2004.

[40] Oppenheimer, M.W., and Doman, D.B., "Reconfigurable Inner Loop Control of a Space Maneuvering Vehicle," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2003-5358, August 2003.

[41] Bolender, M.A., and Doman, D.B., "Nonlinear Control Allocation Using Piecewise Linear Functions," *Journal of Guidance, Control, and Dynamics*, Vol.27, No. 6, pp. 1017-1027, 2004.

[42] Doman, D.B., and Oppenheimer, M.W., "Integrated Adaptive Guidance and Control for Space Access Vehicles, Volume 1: Reconfigurable Control Law for X-40A Approach and Landing," *AFRL IAG&C Technical Report*, Wright-Patterson AFB, OH, 2004.

[43] Schierman, J.D., Ward, D.G., Monaco, J.F., and Hull, J.R., "A Reconfigurable Guidance Approach for Reusable Launch Vehicles," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2001-4429, August 2001.

[44] Cotting, C.M., and Burken, J.J., "Reconfigurable Control Design for the Full X-33 Flight Envelope," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2001-4379, August 2001.

[45] Burken, J.J., and Lu, P., "Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, pp. 482-493, 2001.

[46] Shaffer, Patrick J., *Optimal Trajectory Reconfiguration and Retargeting for the X-33 Reusable Launch Vehicle*, Master's Thesis, Naval Postgraduate School, Monterey, CA, 2004.

[47] Shaffer, P.J., Ross, I.M., Oppenheimer, M.W., Doman, D.B., "Optimal Trajectory Reconfiguration and Retargeting for a Reusable Launch Vehicle," *Proceedings of the 2005 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2005-4168, August 2005.

[48] Schierman, J.D., Hull, J.R., and Ward, D.G., "On-Line Trajectory Command Reshaping for Reusable Launch Vehicles," *Proceedings of the 2003 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2003-5439, August 2003.

[49] Schierman, J.D., Hull, J.R., and Ward, D.G., "Adaptive Guidance with Trajectory Reshaping for Reusable Launch Vehicles," *Proceedings of the 2002 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2002-4458, August 2002.

[50] Allwine, D.A., Fisher, J.E., Strahler, J.A., Lawrence, D.A., Oppenheimer, M.W., and Doman, D.B., "On-Line Trajectory Generation for Hypersonic Vehicles," *Proceedings of the 2005 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2005-6435, August 2005.

[51] Verma, A., Oppenheimer, M.W., and Doman, D.B., "On-Line Adaptive Estimation and Trajectory Reshaping," *Proceedings of the 2005 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2005-6436, August 2005.

[52] Schierman, J.D., Hull, J.R., "In-Flight Entry Trajectory Optimization for Reusable Launch Vehicles," *Proceedings of the 2005 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2005-6434, August 2005.

[53] Carson, J.M., Epstein, M.S., MacMynowski, D.G., and Murray, R.M., "Optimal Nonlinear Guidance with Inner-Loop Feedback for Hypersonic Re-Entry," *Proceedings of the 2006 American Control Conference*, IEEE Paper, June 2006.

[54] Bollino, K.P., Ross, I.M., and Doman, D.B., "Optimal Nonlinear Feedback Guidance for Reentry Vehicles," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2006-6074, August 2006.

[55] Mooij, E., "Model Reference Adaptive Guidance for Re-entry Trajectory Tracking," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2004-4775, August 2004.

[56] Wise, K.A., "Adaptive Control of Flight: Theory, Applications, and Open Problems," *Proceedings of the 2006 American Control Conference*, IEEE Paper, June 2006.

[57] Bryson, A.E., Denham, W.F., Carroll, F.J., and Mikami, K., "Determination of the Lift or Drag Program that Minimizes Re-Entry Heating with Acceleration or Range Constraints using a Steepest Descent Computational Procedure," *Journal of Aerospace Sciences*, Vol. 29, pp. 420-430, 1962.

[58] Speyer, J.L., "Optimization and Control using Perturbation Theory to Find Neighboring Optimum Paths," *SIAM Symposium on Multivariable Linear Control System Theory*, Boston, MA, November 1962.

[59] Hull, D.G., and Speyer, J.L., "Optimal Reentry and Plane-Change Trajectories," *The Journal of Astronautical Sciences*, Vol. XXX, No. 2, pp. 117-130, June 1982.

[60] Pesch, H.J., "Off-Line and On-Line Computation of Optimal Trajectories in the Aerospace Field," Prepared for the 12th Course in *Applied Mathematics in the Aerospace Field* of the International School of Mathematics, Ettore Majorana Centre for Scientific Culture, Sicily, September 1991.

[61] Chou, H.C., Adrema, M.D., and Bowles, J.V., “Near-Optimal Entry Trajectories for Reusable Launch Vehicles,” *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, pp. 983-990, December 1998.

[62] Stengel, R.F., “Optimal Guidance for the Space Shuttle Transition,” *Journal of Spacecraft and Rockets*, Vol. 26, No. 6, pp. 966-972, December 2003.

[63] Windhorst, R., Adrema, M.D., and Bowles, J.V., “Minimum Heating Entry Trajectories for Reusable Launch Vehicles,” *Journal of Spacecraft and Rockets*, Vol. 35, No. 5, pp. 672-682, October 1998.

[64] Betts, J.T., “Survey of Numerical Methods for Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, pp. 193-207, 1998.

[65] Trent, A., Venkataraman, R., and Doman, D., “Trajectory Generation Using a Modified Simple Shooting Method,” *Proc. of the 2004 IEEE Aerospace Conference*, Vol. 4, pp. 2723-2729, March 2004.

[66] Zimmerman, C., Dukeman, G., and Hanson, J., “Automated Method to Compute Orbital Reentry Trajectories with Heating Constraints,” *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, pp. 523-529, August 2003.

[67] Mease, K.D., Teufel, P., Schonenberger, H., Chen, D.T., and Bharadwaj S., “Re-Entry Trajectory Planning for a Reusable Launch Vehicle,” *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 99-4160, August 1999.

[68] Mease, K.D., Chen, D.T., and Schonenberger, H., “Reduced-Order Entry Trajectory Planning for Acceleration Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, pp. 257-266, April 2002.

[69] Saraf, A., Leavitt, J.A., Chen, D.T., and Mease, K.D., “Design and Evaluation of an Acceleration Guidance Algorithm for Entry,” *Journal of Spacecraft and Rockets*, Vol. 41, No. 6, pp. 986-996, December 2004.

[70] Mooij, E., “Parametric Control-Variable Analysis in Support of Re-entry Trajectory Optimisation,” *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper No. 2004-5186, August 2004.

[71] Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.M., “Constrained Model Predictive Control: Stability and Optimality,” *Automatica*, Vol. 36, pp. 789-814, 2000.

[72] Grimm, G., Messina, M.J., Tuna, S.E., and Teel, A.R., “Examples When Nonlinear Model Predictive Control is Nonrobust,” *Automatica*, Vol. 40, pp. 1729-1738, 2004.

[73] Kelley, H.J., "Flight Path Optimization with Multiple Time Scales," *Journal of Aircraft*, Vol. 8, No. 4, pp. 238-240, April 1971.

[74] Van Buren, M.A., and Mease, K.D., "Aerospace Plane Guidance Using Time-Scale Decomposition and Feedback Linearization," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 5, pp. 1166-1174, October 1992.

[75] Shtessel, Y., Hall, C., and Jackson, M., "Reusable Launch Vehicle Control in Multiple-Time-Scale Sliding Modes," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 6, pp. 1013-1020, December 2000.

[76] Lu, P., and Shen, Z., "Unifying Treatment to Control of Nonlinear Systems with Two Timescales," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 5, pp. 975-979, February 2002.

[77] Kitsios, I., and Lygeros, J., "Final Glide-Back Envelope Computation for Reusable Launch Vehicle Using Reachability," *Proc. of the 44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[78] Desai, P., *Improved Collocation Methods with Application to Six-Degree-of-Freedom Trajectory Optimization*, Dissertation, Univ. of Illinois at Urbana-Champaign, March 2005.

[79] Desai, P., "Six-Degree-of-Freedom Trajectory Optimization Using a Two-Timescale Collocation Architecture," AAS/*AIAA Astrodynamics Specialists Conference*, Lake Tahoe, CA, August 2005.

[80] Kirk, D.E., *Optimal Control Theory: An Introduction*, Prentice-Hall, Englewood Cliffs, 1970.

[81] Bryson, A.E., *Dynamic Optimization*, Addison Wesley Longman, Menlo Park, CA, 1999.

[82] Ross, I. M. *Control and Optimization: An Introduction to Principles and Applications, Electronic Edition*, Naval Postgraduate School, Monterey, CA, December 2005.

[83] Josselyn, S. and Ross, I.M., "Rapid Verification Method for the Trajectory Optimization of Reentry Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 3, pp. 505-508, 2002.

[84] Josselyn, S. and Ross, I.M., "Sensitivity Analysis for Rapid Prototyping of Re-Entry Vehicles," *Proceedings of the 2003 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2003-5500, August 2003.

[85] Fahroo, F., Doman, D. and Ngo, A., ``Modeling Issues in Footprint Generation for Reusable Launch Vehicles," *Proceedings of the 2003 IEEE Aerospace Conference*, Big Sky, MT, March 2003.

[86] Fahroo, F., Doman, D. and Ngo, A., ``Footprint Generation for Reusable Launch Vehicles Using a Direct Pseudospectral Method," *Proceedings of the American Control Conference*, Denver, CO, June 2003.

[87] Fahroo, F. and Doman, D., ``A Direct Method for Approach and Landing Trajectory Reshaping with Failure Effect Estimation," *Proceedings of the 2004 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2004-4772, August 2004.

[88] Bollino, K.P., Ross, I.M., and Doman, D.B., "Optimal Nonlinear Feedback Guidance for Reentry Vehicles," *Proceedings of the 2006 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2006-6074, August 2006.

[89] Chou, H-C., Ardrema, M.D., "Near-Optimal Entry Trajectories for Reusable Launch Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, pp. 983-990, December 1998.

[90] Metler, B., and Toupet, O., "Receding Horizon Trajectory Planning with an Environment-Based Cost-to-Go Function," *Proc. of the 44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[91] Johnson, E.J., and Calise, A.J., and Corban, E.J., "Adaptive Guidance and Control for Autonomous Launch Vehicles," *Proc. of the 2001 IEEE Aerospace Conference*, Vol. 6, pp. 2669-2682, March 2001.

[92] JAUS, "JAUS Tutorial," *Joint Architecture for Unmanned Systems*, [online], http://www.jauswg.org/, (accessed October 2006).

[93] Johnson, E.N., Calise, A.J., Curry, M.D., Mease, K.D., and Corban, J.E., "Adaptive Guidance and Control for Autonomous Hypersonic Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 3, pp. 725-737, June 2006.

[94] Lu, P., "Asymptotic Analysis of Quasi-Equilibrium Glide in Lifting Entry Flight," *Proceedings of the 2005 AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper No. 2005-6128, August 2005.

[95] Vinh, Busemann, and Culp, *Optimal Trajectories in Atmospheric Flight (Studies in Astronautics)*, Elsevier, Amsterdam, 1981.

[96] Regan, F.J., and Anandakrishnan, S.M., *Dynamics of Atmospheric Re-Entry,* AIAA Education Series, Washington D.C., 1993.

[97] Zipfel, P.H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA Education Series, AIAA, Reston, VA, 2000.

[98] Anderson, John D., *Introduction to Flight,* 5th Ed., McGraw-Hill Co., New York, 2005.

[99] Etkin, Bernard, *Dynamics of Atmospheric Flight*, John Wiley & Sons, New York, 1972.

[100] Nelson, Robert C., *Flight Stability and Automatic Control*, McGraw-Hill, Inc., New York, 1998.

[101] Harada, M., Bollino, K., and Ross, I.M., “Fuel Optimality of High Altitude Circling for Unmanned Aerial Vehicle,” *2005 JSASS-KSAS Joint International Symposium on Aerospace Engineering*, Nagoya, Japan, October 2005.

[102] Ross, I. M., *User's Manual for DIDO: A MATLAB Application Package for Solving Optimal Control Problems*, Tomlab Optimization, Sweden, February 2004.

[103] Gill, P.E., Murray, W., and Saunders, M.A., *User’s Guide for SNOPT 5.3: A FORTRAN Package for Large-scale Nonlinear Programming*, Technical Report, Stanford Business Software, Inc., Palo Alto, CA, July 1998.

[104] Fahroo, F., and Ross, I.M., “Direct Trajectory Optimization by a Chebyshev Pseudospectral Method,” *Proceedings of the American Control Conference,* Chicago, IL, June 2000.

[105] Fahroo, F., and Ross, I.M., “Direct Trajectory Optimization by a Chebyshev Pseudospectral Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, pp. 160-166, 2002.

[106] Kirk, D.E., *Optimal Control Theory: An Introduction*, Prentice-Hall, Englewood Cliffs, 1970.

[107] Stengel, R.F., *Optimal Control and Estimation*, Dover Publications, Inc., Mineola, NY, 1994.

[108] Bryson, A.E., *Dynamic Optimization*, Addison Wesley Longman, Menlo Park, CA, 1999.

[109] Ross, I.M., *Lecture Notes in Control and Optimization*, Supplemental Lecture Notes for AE3830 and AE4850, Naval Postgraduate School, Monterey, Revised Spring 2005.

[110] Betts, J.T., “Survey of Numerical Methods for Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, pp. 193-207, 1998.

[111] Trent, A., Venkataraman, R., Doman, D., “Trajectory generation using a modified simple shooting method,” *Proceedings of the 2004 IEEE Aerospace Conference*, Vol.4, pp. 2723- 2729, March 2004.

[112] Holsapple, R., Venkataraman, R., Doman, D., “A modified simple shooting method for solving two-point boundary-value problems,” *Proceedings of the 2003 IEEE Aerospace Conference*, Vol.6, pp. 6_2783- 6_2790, March 2003.

[113] Barron, R.L., and Chick, C.M., “Improved Indirect Method for Air-Vehicle Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 3, pp. 643-652, 2006.

[114] Gath, P.F., Well, K.H., “Trajectory optimization using a combination of direct multiple shooting and collocation,” *Proceedings of the 2001 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2001-4047, August 2001.

[115] Betts, J.T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mechanics (SIAM), Philadelphia, PA, 2001.

[116] Bryson, A.E., Ho, Y., *Applied Optimal Control: Optimization, Estimation, and Control,* Rev. Ed., Hemisphere Pub., New York, NY, 1975.

[117] Rardin, R.L., *Optimization in Operations Research*, Prentice Hall, Upper Saddle River, NJ, 1998.

[118] Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., and Mishchenko, E.F., *The Mathematical Theory of Optimal Processes*, Moscow, 1961, translated by K.N. Trirogoff, L.W. Neustadt (Ed.), Interscience, New York, 1962.

[119] Strizzi, J.D., *Pseudospectral Methods for Rapid Computation of Optimal Controls*, Dissertation, Naval Postgraduate School, Monterey, CA, December 2002.

[120] Bazaraa, M.S., Sherali, H.D., and Shetty, C.M., *Nonlinear Programming: Theory and Algorithms*, 2nd Ed., John Wiley & Sons, Inc., New York, 1993.

[121] Salmeron, J., *OA4201: Non-Linear Programming*, Lecture Notes, Naval Postgraduate School, Monterey, 2004.

[122] Ross, I.M., Gong, Q., and Sekhavat, P., “Low Thrust, High-Accuracy Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, accepted for publication, August 2006.

[123] Rea, J.R., *A Legendre Pseudospectral Method for Rapid Optimization of Launch Vehicle Trajectories*, Thesis, MIT, Cambridge, MA, June 2001.

[124] Schoenstadt, A.L., *Numerical Analysis*, Lecture Notes, Naval Postgraduate School, Monterey, 2004.

[125] Gerald, C.F., and Wheatley, P.O., *Applied Numerical Analysis*, 6th Ed., Addison-Wesley, Menlo Park, 1999.

[126] Elnagar, E., Kazemi, M.A., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, pp. 1793-1796, 1995.

[127] Fahroo, F., and Ross, I.M., "A Spectral Patching Method for Direct Trajectory Optimization," *The Journal of the Astronautical Sciences*, Vol. 48, No. 2/3, 2000.

[128] Fahroo, F., and Ross, I.M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, pp. 270-277, 2001.

[129] Boyd, John P., *Chebyshev and Fourier Spectral Methods*, 2nd Ed. (Revised), Dover Publications, Inc., Mineola, New York, 2001.

[130] Strizzi, J., Ross, I.M., and Fahroo, F., "Towards Real-Time Computation of Optimal Controls for Nonlinear Systems," *Proceedings of the 2002 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2002-4945, August 2002.

[131] Ross, I.M., and Fahroo, F., "Legendre Pseudospectral Approximations of Optimal Control Problems," *Lecture Notes in Control and Information Sciences*, Vol. 295, Springer-Verlag, new York, pp. 327-342, 2003.

[132] Ross, I.M., and Fahroo, F., "Pseudospectral Knotting Methods for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, pp. 397-405, 2004.

[133] Ross, I.M., Fahroo, F., and Gong, Q., "A Spectral Algorithm for Pseudospectral Methods in Optimal Control," *Proceedings of the 6th IASTED International Conference on Intelligent Systems and Control*, Honolulu, HI, 2004.

[134] Fornberg, B., *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, Cambridge, 1998.

[135] Ross, I.M., and Fahroo, F., "A Pseudospectral Transformation of the Covectors of Optimal Control Systems," *Proceedings of First IFAC Symposium on System Structure and Control*, Prague, The Czech Republic, August 2001.

[136] Ross, I.M., "A Historical Introduction to the Covector Mapping Principle," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Lake Tahoe, CA, August 2005.

[137] Gong, Q., Ross, I. M., Kang, W. and Fahroo, F., "On the Pseudospectral Covector Mapping Theorem for Nonlinear Optimal Control," to appear in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, December 13-15, 2006.

[138] Ross, I.M., and Fahroo, F., “Convergence of Pseudospectral Discretizations of Optimal Control Problems,” *Proceeding of the 40th IEEE Conference on Decision and Control*, Orlando, FL, December 2001.

[139] Gong, Q., Ross, I.M., Kang, W., and Fahroo, F., “Convergence of pseudospectral methods for constrained nonlinear optimal control problems,” *Proc. 8th IASTED International Conference on Intelligent Systems and Control*, Cambridge, October 2005.

[140] Kang, W., Gong, Q., and Ross, I.M., “Convergence of Pseudospectral Methods for a Class of Discontinuous Optimal Control, *Proc. 44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[141] Gong, Q., Kang, W., and Ross, I.M., “A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems,” *Proc. 2005 IEEE Conference on Control Applications*, Toronto, CN, August 2005.

[142] Gong, Q., Kang, W. and Ross, I. M., “A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems,” *IEEE Transactions on Automatic Control*, Vol. 51, No. 7, pp. 1115-1129, July 2006.

[143] Ross, I.M., and Fahroo, F., “A Direct Method for Solving Nonsmooth Optimal Control Problems,” *Proceedings of the 2002 World Congress of the International Federation of Automatic Control (IFAC)*, Barcelona, Spain, July 2002.

[144] Sekhavat, P., Fleming A., and Ross, I. M., “Time-Optimal Nonlinear Feedback Control for the NPSAT1 Spacecraft,” *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, AIM 2005, 24-28 July 2005 Monterey, CA.

[145] Gong, Q., Ross, I. M. and Kang, W., “A Pseudospectral Observer for Nonlinear Systems,” *Proceedings of the AIAA Guidance, Navigation and Control Conference*, August 2005, San Francisco, CA.

[146] Ross, I.M., Gong, Q., Fahroo, F., and Kang, W., “Practical Stabilization Through Real-Time Optimal Control,” *Proceedings of the American Control Conference*, Minneapolis, MN, June 2006.

[147] Bollino, K., Ross, I.M. and Doman, D., “Optimal Nonlinear Feedback Guidance for Reentry Vehicles,” *AIAA Guidance, Navigation and Control Conference*, Keystone, CO, 2006. AIAA-2006-6074.

[148] Ross, I.M., Sekhavat, P., Gong, Q., and Fleming, A., “Pseudospectral Feedback Control: Foundations, Examples and Experimental Results,” *Proceedings of the 2006 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2004-6354, August 2006.

[149] Fahroo, F. and Ross, I. M., “On Discrete-Time Optimality Conditions for Pseudospectral Methods,” *Proceedings of the AIAA/AAS Astrodynamics Conference*, Keystone, CO, August 2006. AIAA-2006-6304.

[150] Ross, I. M., and Fahroo, F., “Legendre Pseudospectral Approximations of Optimal Control Problems,” *Lecture Notes in Control and Information Sciences*, Vol. 295, Springer-Verlag, New York, 2003.

[151] Ross, I. M. and Fahroo, F., “Issues in the Real-Time Computation of Optimal Control,” *Mathematical and Computer Modeling*, An International Journal, Vol. 43, Issues 9-10, May 2006, pp. 1172-1188. (Special Issue: Optimization and Control for Military Applications).

[152] Ross, I. M. and Fahroo, F., “A Unified Framework for Real-Time Optimal Control,” *Proceedings of the IEEE Conference on Decision and Control*, Maui, December 2003.

[153] Ross, I. M., “A Roadmap for Optimal Control: The Right Way to Commute,” *Annals of the New York Academy of Sciences*, Vol. 1065, New York, NY, December 2005.

[154] Ross, I. M. *Control and Optimization: An Introduction to Principles and Applications, Electronic Edition*, Naval Postgraduate School, Monterey, CA, December 2005.

[155] Mettler, B., and Oliver, T., “Receding Horizon Trajectory Planning with an Environment-Based Cost-to-go Function,” *Proc. 44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[156] Mendy, P.B., *Multiple Satellite Trajectory Optimization*, Master’s Thesis, Naval Postgraduate School, Monterey, CA, 2004.

[157] Fleming, A., “Real-Time Optimal Slew Maneuver Design and Control,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, 2004.

[158] Ross, I. M. “A Case Study in Solving Optimal Control Problems: Formulations, Solutions and Issues”, NASA GRC internal report, March 2004.

[159] Ross, I.M., Gong, Q., and Sekhavat, P., “Low-Thrust, High Accuracy Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, (to appear), 2007.

[160] NASA, “X33 History Project,” http://www.hq.nasa.gov/office/pao/History/x-33/facts_2.htm., (accessed Sep 2006), September 1998.

[161] FAS, “X-33 VentureStar,” http://www.fas.org/spp/guide/usa/launch/x-33.htm., (accessed September 2006), March 1999.

[162] NASA, "The Space Launch Initiative: Technology to pioneer the space frontier," http://www.nasa.gov/centers/marshall/news/background/facts/slifactstext02.htm, (accessed Sep 2006), April 2002.

[163] NASA, "X-33, An Educator's Guide with Activities in Math, Science, and Technology", *NASA EG-1999-08-005-DFRC,* 1999.Shen, Z., *On-board three-dimensional constrained entry flight trajectory optimization*, Dissertation, Iowa State University, Ames, Iowa, 2002.

[164] Kalviste, J., "Spherical Mapping and Analysis of Aircraft Angles for Maneuvering Flight," *Journal of Aircraft*, Vol. 24, No. 8, pp. 523-530, 1987.

[165] Kalviste, J., "Representation and Analysis of Airplane Angles by Spherical Mapping," Northrop Corp., Hawthorne, CA, Report NOR-75-144, October 1975.

[166] Etkin, B., and Reid, L.D., *Dynamics of Flight – Stability and Control*, 3rd Ed., John Wiley & Sons, New York, 1996.

[167] Stengel, Robert, *Flight Dynamics*, Princeton Univ. Press, Princeton, New Jersey, 2004.

[168] McLean, Donald, *Automatic Flight Control Systems,* Prentice Hall, Cambridge, 1990.

[169] Kalviste, J., "Use of Rotary Balance and Forced Oscillation Test Data in a Six Degrees of Freedom Simulation," *Proceedings of the AIAA 9th Atmospheric Flight Mechanics Conference*, AIAA Paper No. 82-1364, August 1982.

[170] Lefferts, E.J., Markley, F.L., and Shuster, M.D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, pp. 417-429, 1982.

[171] Batin, R.H., *An Introduction to the Mathematics and Methods of Astrodynamics*, Rev. Ed., AIAA Education Series, AIAA, Reston, VA, 1999.

[172] NASA, "Aerodynamic Characteristics and Control Effectiveness of the HL-20 Lifting Body Configuration at Mach 10 in Air," *NASA/TM-1999-209357*, Edwards, CA, September 1999.

[173] Cobliegh, B.R., "Development of the X-33 Aerodynamic Uncertainty Model," NASA Dryden Flight Research Center, *NASA/TP-1998-206544*, Edwards, CA, April 1998.

[174] NIMA, "DoD World Geodetic System 1984, Its Definitions and Relationships with Local Geodetic Systems," National Imagery and Mapping Agency Technical Report 8350.2, 3rd Ed., 4 July 1997.

[175] NASA, "The Development of the Joint NASA GSFC and the National Imagery and Mapping Agency (NIMA) Geopotential Model EGM96," Report No. NASA/TP-1998-206861, July 1998.

[176] U.S. Military Specification MIL-F-8785C, 5 November 1980.

[177] U.S. Military Handbook MIL-HDBK-1797, 19 December 1997.

[178] Lewis, L.R., *Rapid Motion Planning and Autonomous Obstacle Avoidance for Unmmaned Vehicles*, Master's Thesis, Naval Postgraduate School, Monterey, CA, 2006.

[179] FAA, "APT Landing Facilities," National Airspace System Resource (NASR), FAA, http://www.faa.gov/ats/ata/ata100/120/stdatafiles.html, (accessed December 2006).

[180] AirNav, http://www.airnav.com/airports/, (accessed November 2006).

[181] Shen, Z., and Lu, P., "Dynamic Lateral Entry Guidance Logic," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, pp. 949-959, 2004.

[182] Phillips, S., "Earth's Atmosphere," *NASA Exploration Database* [online], http://liftoff.msfc.nasa.gov/academy/space/atmosphere.html, (accessed May 2006), December 1995.

[183] Smith, C., "Monthly/Seasonal Climate Composites," NOAA-CIRES Climate Diagnostics Center [online database], http://www.cdc.noaa.gov/cgi-bin/Composites/printpage.pl, (accessed May 2006), October 2005.

[184] Brown, D., Private Communications, Department of Meteorology, Naval Postgraduate School, May 2006.

[185] Creasey, R., Private Communications, Department of Meteorology, Naval Postgraduate School, May 2006.

[186] Bluestein, H.B., *Synoptic-Dynamic Meteorology in Midlatitudes: Volume II: Observations and Theory of Weather Systems (Synoptic-Dynamic Meteorology in Midlatitudes)*, Oxford Univ. Press, New York, 1993.

[187] NOAA, "NCEP/NCAR Reanalysis Project at CDC," NOAA-CIRES Climate Diagnostics Center [online database], http://www.cdc.noaa.gov/cdc/reanalysis/, (accessed May 2006), May 2003.

[188] NOAA/NWS, "The Saffir-Simpson Hurricane Scale," National Hurricane Center -Tropical Prediction Center [online database], http://www.nhc.noaa.gov/aboutsshs.shtml, (accessed May 2006), June 2006.

[189] NOAA, http://www.nnvl.noaa.gov/, NOAA Satellite and Information Service, NESDIS, [on-line], Environmental Visualization Program, accessed November 2006.

[190] Grell, G.A., Dudhia, J., and Stauffer, D.R., “A description of the fifth-generation Penn State/NCEP mesoscale model (MM5), “NCAR Tech. Note NCAR/TN-398+STR, pp. 122, 1995.

[191] Ross, I.M., Gong, Q., and Sekhavat, P., “A Simple Technique for Low-Thrust High-Accuracy Trajectory Optimization,” *16th AAS/AIAA Space Flight Mechanics Conference*, Tampa, FL, Paper No. AAS 06-150, January 2006.

[192] Lu. P., and Vinh, X., “Minimax Optimal Control for Atmospheric Fly-Through Trajectories,” *Journal of Optimization Theory and Applications*, Vol. 57, No. 1, pp. 41-58, April 1998.

[193] Schierman, J.D., Ward, D.G., Hull, J.R., Gandhi, N., Oppenheimer, M.W., and Doman, D.B., “An Approach to Integrated Adaptive Guidance and Control with Flight Test Results,” *Journal of Guidance, Control, and Dynamics*, (submitted), 2004.

[194] Calhoun, P., “An Entry Flight Controls Analysis for a Reusable Launch Vehicle,” *Proc. of the AIAA 38th Aerospace Sciences Meeting*, AIAA Paper No. 2000-1046, January 2000.

[195] Carter, B., *Time-Optimization of High Performance Combat Maneuvers*, Master’s Thesis, Naval Postgraduate School, Monterey, CA, June 2005.

[196] Kang, W., Private Communications, Applied Mathematics, October 2006.

[197] Pan, X., “A General Approach for Multidimensional Smoothing,” *Med Phys.*, Vol. 24, No. 4, pp. 562-570, April 1998.

[198] Moerder, D., *Linear-Quartic Chamfer Splines for Efficient Smooth Modelling of Tabular Data*, NASA-LRC Report, December 1996.

[199] Moon, R. III., *Migration of Real-Time Optimal Control Algorithm: from MATLAB TM to Field Programmable Gate Array (FPGA)*, Master’s Thesis, Naval Postgraduate School, Monterey, CA, 2005.

[200] Ross, I.M., D’Souza, C.N., “Hybrid Optimal Control Framework for Mission Planning,” *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, pp. 686-697, August 2005.

[201] U.S. Army, “Future Combat Systems,” http://www.army.mil\fcs.html, (accessed November 2006).

[202] Shahriar Keshmiri, Private Communications, College of Engineering, Computer Science, and Technology, California State University, Los Angeles, December 2005.

[203] Keshmiri, S., Mirmirani, D., and Colgren, R.D., "Six-DOF Modeling and Simulation of a Generic Hypersonic Vehicle for Conceptual Design Studies," *Proceedings of the AIAA Modeling and Simulation Technology Conference*, AIAA Paper No. 2004-4805, Providence, RI, August 2004.

[204] Keshmiri, and Colgren, R.D., "Development of an Aerodynamic Database for a Generic Hypersonic Air Vehicle," *Proceedings of the 2005 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper No. 2005-6257, August 2005.

[205] ZunZun, "Interactive 2-Dimensional and 3-Dimensional Data Modeling," http://zunzun.com/, accessed November 2006.

[206] Systat, "TableCurve3D", http://www.systat.com/products/TableCurve3D/, accessed November 2006.

THIS PAGE INTENITONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor I. Michael Ross
   Naval Postgraduate School
   Monterey, California

4. Dr. David B. Doman
   Air Force Research Laboratory
   Dayton, Ohio

5. Professor Isaac Kaminer
   Naval Postgraduate School
   Monterey, California

6. Associate Professor Fotis Papoulias
   Naval Postgraduate School
   Monterey, California

7. Professor Wei Kang
   Naval Postgraduate School
   Monterey, California

8. Professor Xiaoping Yun
   Naval Postgraduate School
   Monterey, California

9. Guidance, Navigation, and Control Laboratory
   Attn: Mail Code 71, MAE, Room 208
   Naval Postgraduate School
   Monterey, California